

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
Факультет кібербезпеки, комп'ютерної та програмної інженерії  
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач випускової кафедри  
\_\_\_\_\_ А.С. Савченко  
« \_\_\_\_ » \_\_\_\_\_ 2021\_р

# ДИПЛОМНИЙ ПРОЕКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»

**Тема:** «Веб інтерфейс для водіїв децентралізованої логістичної компанії»

**Виконавець:** студент УС-411 Коровін Дмитро Олегович  
(студент, група, прізвище, ім'я, по батькові)

**Керівник:** д.т.н. професор Зіатдінов Юрій Кашафович  
(науковий ступень, вчене звання, прізвище, ім'я, по батькові)

**Нормоконтролер:** ст. викл. Шевченко О.П.  
(П.І.Б.) (підпис)

КИЇВ 2021

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

Освітній ступінь: Бакалавр

Галузь знань, спеціальність, спеціалізація: 12 “Інформаційні технології”,  
122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

А.С. Савченко

“\_\_\_\_\_” \_\_\_\_\_ 2021 р.

## ЗАВДАННЯ

**на виконання дипломного проекту студента**

Коровін Дмитро Олегович

(прізвище, ім'я, по батькові)

1. Тема проекту: «Веб інтерфейс для водіїв децентралізованої логістичної компанії»  
затверджена наказом ректора № 636/ст. від 22.04.2021р.
2. Термін виконання роботи: з 10.05.2021 по 12.06.2021р.
3. Вихідні дані до роботи: розробка структури інтерфейсу веб сервісу для водіїв децентралізованої логістичної компанії.
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): вступ, аналітичний огляд і постановка завдання, розгляд завдання створення інтерфейсів веб сервісів, дослідження технологій та засобів, розробка програмного продукту інтерфейсу веб сервісу для водіїв децентралізованої логістичної компанії, оцінка якості технології, висновки.
5. Перелік обов'язкового графічного матеріалу: загальний перелік існуючих систем та обробка інформації створеним програмним продуктом. Використання структури проблематики питань створення інтерфесйів.

## КАЛЕНДАРНИЙ ПЛАН-ГРАФІК

| №<br>п/п | Етапи виконання дипломної роботи   | Термін<br>виконання етапів | Примітка |
|----------|--|----------------------------|----------|
| 1.       | Аналіз літератури та джерел за темою дипломного проекту.                           | 11.05.2021 -<br>12.05.2021 |          |
| 2.       | Розроблення та затвердження плану дипломного проекту.                              | 13.05.2021                 |          |
| 3.       | Опрацювання інформації за тематикою.   | 14.05.2021                 |          |
| 4.       | Розробка розділу 1.  | 15.05.2021 -<br>18.05.2021 |          |
| 5.       | Розробка розділу 2.  | 05.06.2021 -<br>08.06.2021 |          |
| 6.       | Розробка розділу 3.  | 09.06.2021 -<br>10.06.2021 |          |
| 7.       | Написання пояснювальної записки. Підготовка графічного демонстраційного матеріалу. | 11.06.2021 -<br>12.06.2021 |          |

Студент

( *Коровін Д.О.* )

Керівник дипломної роботи

( *Зіатдінов Ю.К.* )

## **РЕФЕРАТ**

Пояснювальна записка до дипломної роботи «веб інтерфейс для водіїв децентралізованої логістичної компанії»: 54 с., 26 рис., 11 літературних джерел.

Об'єкт дослідження: графічні веб інтерфейси користувача.

Мета роботи: створення графічного веб інтерфейсу для водіїв децентралізованої логістичної компанії з метою раціоналізації контролю робочого часу працівників компанії та зручності обробки та надання інформації водіями.

Методи дослідження: аналіз сучасних засобів створення графічних веб інтерфейсів користувача, аналіз головного процесу логістичної компанії – процедура перевезення вантажу.

Отримані результати та їх новизна: графічний інтерфейс, структура інтерфейсу є раціональною, доступною для розширення та інтеграції нового функціоналу.

**ГРАФІЧНИЙ ІНТЕРФЕЙС, СТРУКТУРА КОМПОНЕНТІВ ІНТЕРФЕЙСУ,  
ВЕБ СЕРВІС, ЛОГІСТИКА**

## ЗМІСТ

|   |    |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ .....  | 7  |
| ВСТУП.....  | 8  |
| РОЗДІЛ 1. РОЗГЛЯД ЛОГІСТИЧНОЇ ДІЯЛЬНОСТІ .....  | 10 |
| 1.1. Загальні положення .....   | 10 |
| 1.2. Транспортна логістика .....  | 14 |
| 1.3. Головні сутності системи логістичної компанії.....   | 16 |
| 1.4. Апаратне та програмне забезпечення логістичних компаній.....   | 17 |
| РОЗДІЛ 2. ІНТЕРФЕЙСИ ВЗАЄМОДІЇ ПРАЦІВНИКІВ ЛОГІСТИЧНОЇ КОМПАНІЇ   | 21 |
| 2.1. Загальне відображення інтерфейсів системи .....  | 21 |
| 2.2. Загальна структура інтерфейсу веб сервісу водіїв децентралізованої логістичної компанії. Призначення та основні функції..... | 25 |
| 2.3. Діаграма атрибутів класів елементів інтерфейсу.....  | 26 |
| 2.4. Огляд методів та алгоритмів створення інтерфейсу водіїв .....  | 29 |
| 2.4.1. Алгоритм оптимізації швидкодії інтерфейсу бібліотекою React.....   | 30 |
| 2.4.2. Алгоритм формування ієрархії даних за допомогою бібліотеки Redux .....   | 31 |
| 2.4.3. Метод покращення взаємодії з React за допомогою NextJs.....  | 34 |
| РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ .....  | 35 |
| 3.1. Встановлення залежностей проекту.....  | 35 |
| 3.2. Створення колекцій бази даних .....  | 39 |
| 3.3. Налаштування оновлення повідомлень чату в режимі реального часу.....   | 41 |
| 3.4. Основні елементи веб інтерфейсу .....  | 42 |
| 3.4.1. Компонент замовлень призначених водієві .....  | 43 |
| 3.4.2. Компонент ведення план-графіку діяльності водія .....  | 48 |
| 3.4.3. Компонент мапи .....   | 50 |
| 3.4.4. Компонент чату .....   | 52 |
| ВИСНОВКИ.....   | 53 |
| СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....  | 54 |

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ**

|      |   |                                 |
|------|---|---------------------------------|
| EOM  | – | електрона обчислювальна машина. |
| APT  | – | Advanced Packaging Tool.        |
| NPM  | – | Node Package Manager.           |
| МДП  | – | міжнародні дорожні перевезення. |
| HTML | – | Hyper Text Markup Language.     |
| CSS  | – | Cascading Style Sheets.         |
| DOM  | – | Document Object Model.          |

## ВСТУП

Глобалізація та посилення конкуренції на світовому транспортному ринку призвели до необхідності застосування принципово нових підходів до формування міжнародних, регіональних і галузевих економічних відносин.

Авіап перевезення є найнадійнішими та найшвидшими у світі. Але на даний час аеропорти великих країн несуть сильне навантаження. Інколи навіть пасажирським літакам доводиться кружляти навколо аеропорту близько 40 хвилин, очікуючи свою чергу на приземлення. А для будівництва нових аеропортів та великих логістичних центрів не вистачає земель.

Досвід зарубіжжя показує, що більшість операцій перетворення матеріального потоку та товарообігу зовнішньої торгівлі здійснюються у логістичних центрах. Підприємства економічно розвинених країн світу формують власні логістичні центри або користуються послугами, які вони надають. Проведені нами дослідження показують, що логістичні центри для різних галузей в Україні ще не сформовані. Такий фактор призупиняє розвиток економіки країни та економіки окремих регіонів. Незадовільний стан розвитку логістики в регіональному аспекті негативно впливає на перебіг таких економічних процесів як постачання, швидкість перетворення матеріальних та інших логістичних потоків, покращення якісних характеристик логістичних операцій, доставка готової продукції, товарообіг різних галузей, експорт-імпорт країни, транзит. За таких умов знижується приплив інвесторів в окремих регіонах, галузях.

Саме тому необхідність створення логістичних децентралізованих компаній з власними логістичними центрами, які надають швидкий та ефективний процес розподілення вантажів та товарів різного призначення у будь-які точки світу з раціональним використанням часових та людських ресурсів стає критичною.

Раціональність та швидкодію взаємодії інстанцій логістичної компанії забезпечується якістю обробки даних, які формуються адміністрацією, керівництвом інстанцій та працівниками вантажних складів та терміналів і що не мало важливо

водіями транспортних засобів, які виконують доставки замовлень до кінцевого користувача.

**Актуальність теми дипломної роботи обумовлена:**

- необхідністю здійснення структурних змін, спрямованих на інтеграцію авіатransпортної галузі України до міжнародної транспортно-логістичної системи;
- пошуком можливостей підвищення ефективності діяльності вітчизняних авіатransпортних підприємств;
- впровадженням у світову практику оптимальних схем доставки вантажу за логістичними принципами «від дверей до дверей» та «точно в термін»;
- сучасними вимогами щодо координації і синхронізації процесів обробки вантажів для скорочення часу виконання транспортно-розподільчих процесів;
- зростанням потреб сучасної авіаційної клієнтури щодо спектру послуг та якості обслуговування вантажів;
- впровадженням нових технологій наземного обслуговування вантажів в аеропортах.

Таким чином, все вище сказане обумовило вибір напрямку наукового дослідження і свідчить про його актуальність та дає підстави стверджувати, що на сьогоднішній день практично відсутні роботи, в яких комплексно і безпосередньо досліджуються особливості створення механізму формування міжнародного транспортно-логістичного центру на базі аеропорту для вирішення загального питання реалізації концепції інтегрованої логістики на повітряному транспорті та подальшого розповсюдження наземним транспортом.



# РОЗДІЛ 1

## РОЗГЛЯД ЛОГІСТИЧНОЇ ДІЯЛЬНОСТІ

### 1.1. Загальні положення

У найширшому сенсі логістикою називають будь-які процеси пов'язані з транспортуванням, зберіганням та обробкою будь-яких предметів. Термін «логістика» був вперше застосований у військових операціях і стосувався насамперед для процесів забезпечення армійських підрозділів. Останніми роками, однак, термін набув широкого вжитку в бізнесі й тепер асоціюється саме з корпоративною лексикою.

Основним об'єктом управління логістики, як господарської діяльності, є матеріальний потік, що проходить по логістичному ланцюгу, починаючи від первинного джерела сировини через усі проміжні процеси аж до постачання готової продукції до кінцевого споживача.

Зміст концепції логістики полягає в тому, що вона допомагає формувати науково-практичні основи розробки і реалізації рекомендацій, які спрямовані на підвищення ефективності управління матеріальними і супутніми їм інформаційними і фінансовими потоками, виходячи з логістичного середовища, що сформувалося на ринку, а також на відпрацювання нових підходів до управління логістичними активностями в усіх ланках логістичної системи, ланцюга поставок.

Сучасна концепція логістики базується на наступних основних положеннях, а саме: на обліку логістичних витрат у межах логістичної системи (або логістичного ланцюга), гуманізації технологічних процесів і створенні сучасних умов праці для логістичного персоналу, розвитку і впровадженні логістичного сервісу.

| Кафедра КІТ (47) |                |  |  | НАУ 21 14 92 000 ПЗ               |         |       |         |
|------------------|----------------|--|--|-----------------------------------|---------|-------|---------|
| Виконав          | Коровін Д.О.   |  |  | РОЗГЛЯД ЛОГІСТИЧНОЇ<br>ДІЯЛЬНОСТІ | Літера  | Аркуш | Аркушів |
| Керівник         | Зіатдінов Ю.К. |  |  |                                   |         | 9     | 54      |
| Консульт.        |                |  |  |                                   | 411 122 |       |         |
| Н-котрол.        | Шевченко О.П.  |  |  |                                   |         |       |         |
| Зав. каф.        | Савченко А.С.  |  |  |                                   |         |       |         |

*Логістика* – це гармонізація інтересів учасників процесу переміщення продукції, форма оптимізації ринкових зв'язків, тобто вдосконалення управління матеріальними і пов'язаними з ними інформаційними та фінансовими потоками на шляху від первинного джерела сировини до кінцевого споживача готової продукції на основі системного підходу й економічних компромісів для отримання синергічного ефекту.

Найбільш узагальнюючим є концептуальний підхід, за яким *логістика* – це:

- сукупність різних видів діяльності з метою одержання з мінімальними витратами необхідної кількості продукції в установлений час та в установленому місці, в якому існує конкретна потреба в даній продукції;
- деяка система, розроблена для кожного підприємства з метою оптимального, з точки зору отримання прибутку, прискорення руху матеріальних ресурсів і товарів усередині та поза підприємством, починаючи від закупівлі сировини та матеріалів, проходження їх через виробництво і закінчуючи постачанням готових виробів споживачам, включаючи інформаційну систему, яка зв'язує ці задачі;
- наука управління матеріальними потоками від первинного джерела до кінцевого споживача з мінімальними витратами, пов'язаними з товаропросуванням і відносним потоком інформації;
- способи і методи координування відносин підприємства з партнерами, засіб координування поставленого ринком попиту і виставленої підприємством пропозиції;
- спосіб організації діяльності підприємства, який дозволяє поєднати зусилля різних одиниць, виробляючих товари і послуги, з метою оптимізації фінансових, матеріальних і трудових ресурсів, які використовує підприємство для реалізації своїх економічних цілей;
- науково-практичний напрям господарювання, який закладається в ефективному управлінні матеріальними і пов'язаними з ними інформаційними та фінансовими потоками в сферах виробництва та обігу;

- теорія і практика управління матеріальними і пов'язаними з ними інформаційними потоками;
- міждисциплінарний науковий напрям, безпосередньо пов'язаний з пошуком нових можливостей підвищення ефективності матеріальних потоків;
- напрям господарської діяльності, який закладається в управлінні матеріальними потоками в сферах виробництва і обігу;
- наука про планування, контроль і управління транспортуванням, складуванням та іншими матеріальними і нематеріальними операціями, здійснюваними в процесі доведення сировини і матеріалів до виробничого підприємства, внутрізаводської переробки сировини, матеріалів та напівфабрикатів, доведення готової продукції до споживача у відповідності з його інтересами і вимогами, а також передачі, зберігання та обробки відповідної інформації;
- наука про процес фізичного розподілу продукції в просторі і часу;
- наука про взаємозв'язок і взаємодію постачання зі збутом і транспортом;
- комплексний напрям в економіці, який охоплює проблеми управління матеріальними потоками;
- наука про раціональну організацію виробництва і розподілення, яка в комплексі вивчає постачання, збут і розподілення засобів виробництва.

Більш конкретним до розуміння логістики є функціональний підхід. Він зумовлює її визначення з погляду виконуючих підприємством логістичних функцій, згідно з таким підходом логістика – це:

- процес дослідження і прогнозування ринку, планування виробництва, закупівлі сировини, матеріалів та обладнання, включаючи контроль за запасами і ряд послідовних товаро-просувних операцій, вивчення технології обслуговування покупців;
- переміщення та зберігання сировини, напівфабрикатів і готових виробів в господарському обігу з моменту сплати грошей постачальнику до моменту одержання грошей за постачання кінцевої продукції споживачу;

- процес планування, контролю та управління формуванням матеріального потоку, його складуванням та інтегрованою інформацією від місця виготовлення до місця споживання з метою пристосування до потреб споживача;
- процес управління просуванням і зберіганням сировини, компонентів та готової продукції в господарському обігу з моменту сплати грошей постачальникам до моменту отримання грошей за доставку готової продукції споживачу.

Однак всі наведенні тлумачення не суперечать концептуальній сутності логістики, а їх численність є цілком закономірним явищем, яке часто супроводжує становлення і розвиток нових науково-практичних напрямків.

При цьому не важко помітити, що основним об'єктом дослідження, управління й оптимізації вважається саме матеріальний потік. Пізніше в сферу інтересів логістики потрапили інформаційні та фінансові потоки, які супроводжують матеріальний, а зовсім недавно - потоки послуг. Сьогодні здійснюються спроби подальшого розширення сфери застосування логістики шляхом виділення як об'єктів її дослідження енергетичних, трудових та інших потоків, які присутні в економічних системах.

У даному навчальному курсі логістика розглядається як теорія і практика управління матеріальними і пов'язаними з ними інформаційними і фінансовими потоками.

Новизна логістичного підходу в управлінні ресурсами полягає в зміні пріоритетів господарської діяльності. Головну роль відіграє не продукт, а процес у формі потоку (матеріального, інформаційного, фінансового тощо). Управління поточковими процесами, їх перетворення й інтеграція є новою формою управління, що перевершує традиційні як за рівнем творчого потенціалу, так і за ефективністю кінцевих результатів. Оптимізація поточкових процесів в економіці стала можливою лише завдяки переорієнтації з кількісних критеріїв оцінки господарської діяльності на якісні.

Як науковий напрямок логістика і далі розширює межі свого застосування. На сьогодні вона виділилася в спеціальну дисципліну, тісно пов'язану з математикою, кібернетикою, статистикою та певними економічними науками.

Теоретичні положення і конкретні рекомендації логістики активно впроваджуються в практичну діяльність фірм і компаній у багатьох країнах. У прикладній сфері зворотна віддача виявляється у відчутному економічному ефекті, такому як скорочення витрат і часу в сферах виробництва і обігу.

До логістики як наукової основи управління поточковими процесами звертаються не тільки у промисловості, торгівлі і на транспорті, але також у сфері послуг, банківській і страховій справі, організації після продажного сервісу, у комунальному господарстві, у сфері туризму та інших галузях діяльності.

## **1.2. Транспортна логістика**

*Транспортна логістика* - це система по організації доставки, а саме переміщення будь-яких матеріальних предметів або речовин з однієї точки в іншу за оптимальним маршрутом. Одна із основоположних напрямків науки про управління інформаційними і матеріальними потоками у процесі руху товарів.

Оптимальним вважається той маршрут, по якому можливо доставити логістичний об'єкт в найкоротші терміни (або передбачені терміни) з мінімальними витратами, а також з мінімальною шкодою для об'єкта доставки.

Шкодою для об'єкта доставки вважається негативний вплив на логістичний об'єкт як з боку зовнішніх чинників (умови перевезення), так і з боку тимчасового фактору при доставці об'єктів, які підпадають під дану категорію.

Задачі:

- вибір типу транспортного засобу;
- вибір виду транспортного засобу;
- спільне планування транспортних процесів із складськими та виробничими операціями;
- спільне планування транспортних процесів на різних видах транспорту;

- забезпечення технологічної єдності транспортно-складського процесу;
- визначення раціональних маршрутів постачання.

**Транспортування** (англ. Transportation) — процес переміщення людей, вантажів, сигналів та інформації з одного місця в інше. В логістиці транспортування розглядається, як процес переміщення людей та вантажів, або перевезення. Розглядають транспортування технологічне (внутрішнє) та зовнішнє. Також, для опису процесу переміщення використовується термін перевезення.

Внутрішнє переміщення товарів здійснюється в процесі виробництва та складування, як правило, в межах одного виробничого майданчика на незначні відстані. Зовнішнє здійснюється при перевезенні, як правило, на далекі відстані між різними суб'єктами господарювання, або між віддаленими підрозділами одного підприємства.

Організація перевезення є важливою функцією. Існує декілька підходів до організації перевезень. Серед них можемо назвати два основних: децентралізоване та централізоване транспортування:

За децентралізованого перевезення кожен підрозділ компанії самостійно, без узгодження й координації з іншими займається переміщенням необхідних вантажів.

За централізованого перевезення, тільки вповноважений транспортний підрозділ має право здійснювати операції із перевезень. Інші підрозділи компанії виставляють замовлення до перевезення.

Транспортування може здійснюватися різними видами транспорту:

- автомобільним транспортом;
- залізничним транспортом;
- морським транспортом;
- річковим транспортом;
- повітряним транспортом;
- трубопровідним транспортом.

У процесі прийняття рішення про перевезення найманим чи власним транспортом оцінюється наступне:

- загальні витрати;

- рівень контролю;
- рівень сервісу;
- гнучкість.

Після вибору виду транспорту й перевізника встановлюється процедура оцінки якості транспортних послуг, яка включає в себе:

- аналіз витрат;
- облік пошкоджень продукції;
- оцінка своєчасності доставки.

В автомобільних перевезеннях важливу роль відіграють транспортно-експедиційні компанії. Міжнародні перевезення здійснюються відповідно до Митної конвенції про міжнародні перевезення із застосуванням книжки МДП.

**Складування** — це дії з утримання матеріальних цінностей на складі. Належить до логістичної підсистеми зберігання, яке є ширшим поняттям, оскільки охоплює інші операції з переробки товарів на складах: розвантаження, приймання, сортування, постановка на власне зберігання, виймання, комплектація, очікування, завантаження (логістичне), документальне оформлення всіх операцій.

### 1.3. Головні сутності системи логістичної компанії

**Транспортно-логістичні комплекси** являють собою багатофункціональні високотехнологічні виробничі об'єкти, координуючі взаємодія різних видів виробничих підприємств на основі комплектації, зберігання і перевезення вантажів, розвитку сервісних комерційно-ділових та інформаційних послуг Їх можна вважати складовими елементами будь-якого сучасного виробництва Вони відрізняються розмірами територій та обсягами послуг.

Основна функція сучасного транспортно-логістичного центру - надання складських приміщень з цілим комплексом додаткових послуг:

- навантаження-розвантаження, маркування та зберігання вантажів;
- організація повітряних вантажоперевезень;
- технічне обслуговування транспортних засобів;

- митне оформлення;
- брокерські послуги;
- інформаційні послуги.

Сучасний логістичний центр найчастіше представляє собою багатофункціональний комплекс, що займає кілька тисяч кв.м., тому його проект може включати планування:

- складських будівель;
- розворотних майданчиків і стоянок літаків;
- електропідстанції;
- котельні (автономний енерговузол);
- ВЗУ;
- офісних приміщень;
- АБК (адміністративно-побутовий комплекс);
- побутових приміщень (роздягальні, душові, кімнати для відпочинку, їдальні);
- поста охорони.

#### **1.4. Апаратне та програмне забезпечення логістичних компаній**

Протягом останніх років бурхливо розвиваються засновані на інформатиці так називані нові логістичні технології. Інформаційні системи займають у цих технологіях центральне положення. Розвиток логістики в розвинутих країнах не в останню чергу стимулюються необхідністю швидкої реакції виробників на кон'юнктуру ринку, прагненням у короткий час адаптуватися в ситуаціях, що змінюються.

Центральною ідеєю логістики є планування, керування і контроль підприємницької діяльності, усіх матеріальних і інформаційних потоків зв'язаних з цією діяльністю. Інформаційні системи в логістики припускають швидку адекватну реакцію на вимогу ринку, спостереженням за часом доставки, оптимізацію функцій у ланцюгах доставки і постачання й інше.



Але і тут виникають труднощі і проблеми створення інформаційних систем на підприємстві. Однією з перших проблем – відсутність збору інформації на підприємствах. В основному інформація носить не точний, не оперативний і не спадкоємний характер. Найчастіше компанії терплять крах через несвоєчасний, або недостовірної отриманої інформації.

Що і говорити в нашій країні подібні проблеми ведуть просто до хаосу в економіці. Також слабкий розвиток комунікаційних мереж за структурою і технічним рівнем для інформаційних систем обслуговуючих ЕОМ і відсутність інформаційної взаємодії між постачальниками-виробниками і покупцями-споживачами стають наступною проблемою створення добре налагоджених логістичних інформаційних систем.

Що ж можна зробити з цього приводу? Можна створити галузеві і внутрівиробничі центри, що будуть керувати одночасно інформаційними і матеріальними потоками на підприємстві. Також можна створити логістичну інформаційну систему в сфері підготовки вантажів до перевезення з застосуванням електронних перевізних документів у внутрішніх і прямим міжнародних повідомленнях (про це ми поговоримо пізніше і докладніше). І нарешті-те налагодимо логістичну взаємодія постачальників і покупців транспорту і виробництва, створити гнучкі методи керування з орієнтацією на економію витрат ресурсів і енергії.

Наступна проблема, на мою думку є проблема в сфері зовнішньоекономічної діяльності підприємства, при подоланні митних бар'єрів, а особливо для держав, що знаходяться в безпосередній близькості один від одного і які практично щодня мають зв'язок.

Для того щоб полегшити відносини в цій області потрібно ввести єдину міжнаціональну комунікаційну інформаційну систему. Вона буде призначена для передачі інформації про матеріальні потоки і контроль за їхнім рухом.

Ця інформаційна система буде поєднувати комунікаційні системи багатьох країн і в зв'язку з цим вона зменшить час перебування вантажів на прикордонних станціях і зв'язані з цим витрати. Ця система повинна бути постійно відкрита для

користувачів із приводу обміну даними, а також для користування цією системою потрібно –створити загальну мову, як єдину.

Система буде мати в наявності самостійну комунікаційну мережу і цю мережу, не буде залежати від інших державних інформаційних систем. Також вона повинна забезпечувати цілодобовий обмін інформацією між користувачами. Крім того вона буде розвиватися при збільшенні потоків інформації і числа абонентів.

При інформаційному забезпеченні всіх підприємств можна буде замінити деякі паперові документи – електронними аналогами, що дозволяє синхронізувати рух матеріальних і інформаційних потоків, а також скоротяться витрати при підготовці паперових документів. Однак це теж буде здійснити не завжди і не так просто як здається.

У деяких випадках виникають зайві юридичні складності. Але для того, щоб усе це здійснити потрібно впливати прогресивним технологіям, таким як: канали зв'язку, побудовані з використанням лазерів, засобів космічного зв'язку тощо, а тому що це все дуже, дуже дорого, це теж стає вагомою проблемою для багатьох підприємств.

**Інформаційні системи забезпечують підготовку, введення, збереження, обробку, контроль і передачу даних.** Вони відрізняються ієрархічною структурою. Ступінь їхньої автоматизації звичайно відносно висока. Інформаційні системи бувають реалізовані як мережа взаємозалежних обчислювальних машин різної величини й абонентських пунктів (терміналів). Їхньої підсистеми виконують функції на різних рівнях керування, як правило, використовуючи загальний банк даних. Інформацію усе більш високого рівня одержують стиском даних з детальної бази більш низьких рівнів. Детальні дані передаються на більш високий рівень тільки у випадку значних відхилень від необхідного чи очікуваного стану.

Логістичні інформаційні системи являють собою відповідні інформаційні мережі, що починаються з денних вимог замовників (що представляють чисто стохастичну величину), що поширюються через розподіл і виробництво до постачальників.

**Інформаційні системи для прийняття довгострокових рішень про структури і стратегії** (так названі планові системи). Вони служать головним чином для створення й оптимізації ланок логістичного ланцюжка. Для планових систем характерна пакетна обробка задач.

**Інформаційні системи для прийняття рішень на середньострокову і короткострокову перспективу** (так називані диспозитивні чи диспетчерські системи). Вони спрямовані на забезпечення налагодженої роботи логістичних систем. Мова йде, наприклад, про розпорядження (диспозиції) внутрішньозаводським транспортом, запасами готової продукції, забезпеченні матеріалами і підрядними постачаннями, запуску замовлень у виробництво. Деякі задачі можуть бути оброблені в пакетному режимі, інші вимагають інтерактивної обробки (online) через необхідність використовувати як можна більш актуальні дані. Диспозитивна система підготовляє усі вихідні дані для прийняття рішень і фіксує актуальний стан системи в базі даних.

**Інформаційні системи для виконання повсякденних справ** (так називані виконавчі системи). Вони використовуються головним чином на адміністративному й оперативному рівнях керування, але іноді містять також деякі елементи короткострокової диспозиції. Особливо важливі для цих систем швидкість обробки і фіксування фізичного стану без запізнювання (тобто актуальність усіх даних), тому вони в більшості випадків працюють у режимі online. Мова йде, наприклад, про керування складами й обліку запасів, підготовці відправлення, оперативному керуванні виробництвом, керуванні автоматизованим устаткуванням. Керування процесами й устаткуванням вимагає інтеграції інформаційних систем комерційного характеру і систем керування автоматикою.

**Для побудови логістичних інформаційних систем на базі ЕОМ важливі наступні принципи:**

- потрібно прагнути до модульної структури систем як в апаратному устаткуванні, так і в програмному забезпеченні;
- треба забезпечити можливість поетапного створення системи;
- дуже важливим є чітке встановлення місць стику;

- потрібно забезпечити гнучкість системи з погляду специфічних вимог конкретного застосування;
- ведучу роль грає прийнятність системи для користувача діалогу "людина-машина".

**Стратегічне планування інформаційної системи включає наступні кроки:**

- визначення підрозділів підприємства, що будуть включені в інтегровану інформаційну систему (також з урахуванням перспективи);
- грубий проект функціональних областей інформаційної системи і співвідношень між ними;
- визначення важливих для роботи підприємства об'єктів (замовники, постачальники матеріалів, деталей і т. п.) і їхнє відображення в інформаційній системі (це найбільш складна задача стратегічного планування, тісно зв'язана з попереднім кроком);
- визначення можливостей використання функціональних областей системи в різних підрозділах підприємства й оцінка очікуваного ефекту;
- установлення правил для архітектури і технічної реалізації підсистем і з'єднуючих ланок, створюваних власними силами;
- установлення загальних, незалежних від функцій правил і форматів для передачі даних між функціональними областями інформаційної системи;
- установлення параметрів для обчислювальної техніки (апаратне устаткування, операційна система, система керування даними, ієрархічні рівні ЕОМ, технічні методи передачі);
- розробка проекту реалізації (пріоритети, терміни і т. д.).

Якщо в інформаційній системі здійснюється автоматизована обробка інформації, то технічне забезпечення містить у собі електронну обчислювальну техніку і засоби зв'язку між собою. Основною частиною технічного забезпечення в цьому випадку є ЕОМ.

Одним з основних блоків сучасної електронної обчислювальної машини є процесор – пристрій, що здійснює запрограмовану обробку даних. Розвиток електроніки дозволило робити процесори дуже невеликих розмірів, що володіють

значною швидкістю й обсягом пам'яті. ЕОМ, виконану на базі мікропроцесорів, відносять до мікро-ЕОМ.

## РОЗДІЛ 2

### ІНТЕРФЕЙСИ ВЗАЄМОДІЇ ПРАЦІВНИКІВ ЛОГІСТИЧНОЇ КОМПАНІЇ

#### 2.1. Загальне відображення інтерфейсів системи

Запорукою економічного успіху будь-якої країни, якщо брати глобально чи компанії – локально, значно залежить від того, наскільки раціонально вони використовують найбільш цінний ресурс – час. Наскільки швидко система отримує актуальні данні, реагує на них та приймає відповідні рішення з урахуванням усіх критеріїв впливу на поточні процеси. Логістичні компанії не є винятком. Ключовим аспектом успішного існування таких компаній є контроль у режимі реального часу усіх обсягів товару та вантажу, а також, що не менш важливо, контроль роботи працівників відповідних відділів.

Подібний контроль неможливий без інтеграції інформаційних технологій до ключових систем логістичної компанії, адже вони забезпечують найефективнішу комунікацію між різними відділами та надає коректний контроль актуальності даних, а також визначає ієрархічну структуру цих даних. Основні проблеми виникають коли розмитнений вантаж необхідно розвести наземними транспортними засобами до кінцевих споживачів, адже ніхто не застрахований від випадкових ситуацій, які можуть стати результатом створення недостовірних даних, потрапити до головної системи та призвести до збою усього підприємства.

| Кафедра КІТ (47) |                |  |  | НАУ 21 14 92 000 ПЗ   |         |       |         |
|------------------|----------------|--|--|---|---------|-------|---------|
| Виконав          | Коровін Д.О.   |  |  | ІНТЕРФЕЙСИ ВЗАЄМОДІЇ<br>ПРАЦІВНИКІВ ЛОГІСТИЧНОЇ<br>КОМПАНІЇ | Літера  | Аркуш | Аркушів |
| Керівник         | Зіатдінов Ю.К. |  |  |   |         | 22    | 54      |
| Консульт.        |                |  |  |   | 411 122 |       |         |
| Н-котрол.        | Шевченко О.П.  |  |  |   |         |       |         |
| Зав. каф.        | Савченко А.С.  |  |  |   |         |       |         |

Аби уникнути подібних ситуацій працівники кожного з відділів компанії повинні бути оснащені відповідними електронними пристроями для взаємодії з якими необхідно використовувати інтерфейси та сервіси, які надають чітке розуміння того, які данні працівник повинен надати, в який проміжок часу та в якому об'ємі, щоб керуючі механізми верхніх рівнів відповідним чином їх обробили та надали актуальну інформацію до працівників іншого відділу, щоб вони змогли приступити до виконання власних робочих обов'язків та продовжити процес обробки інформації про поточний вантаж.

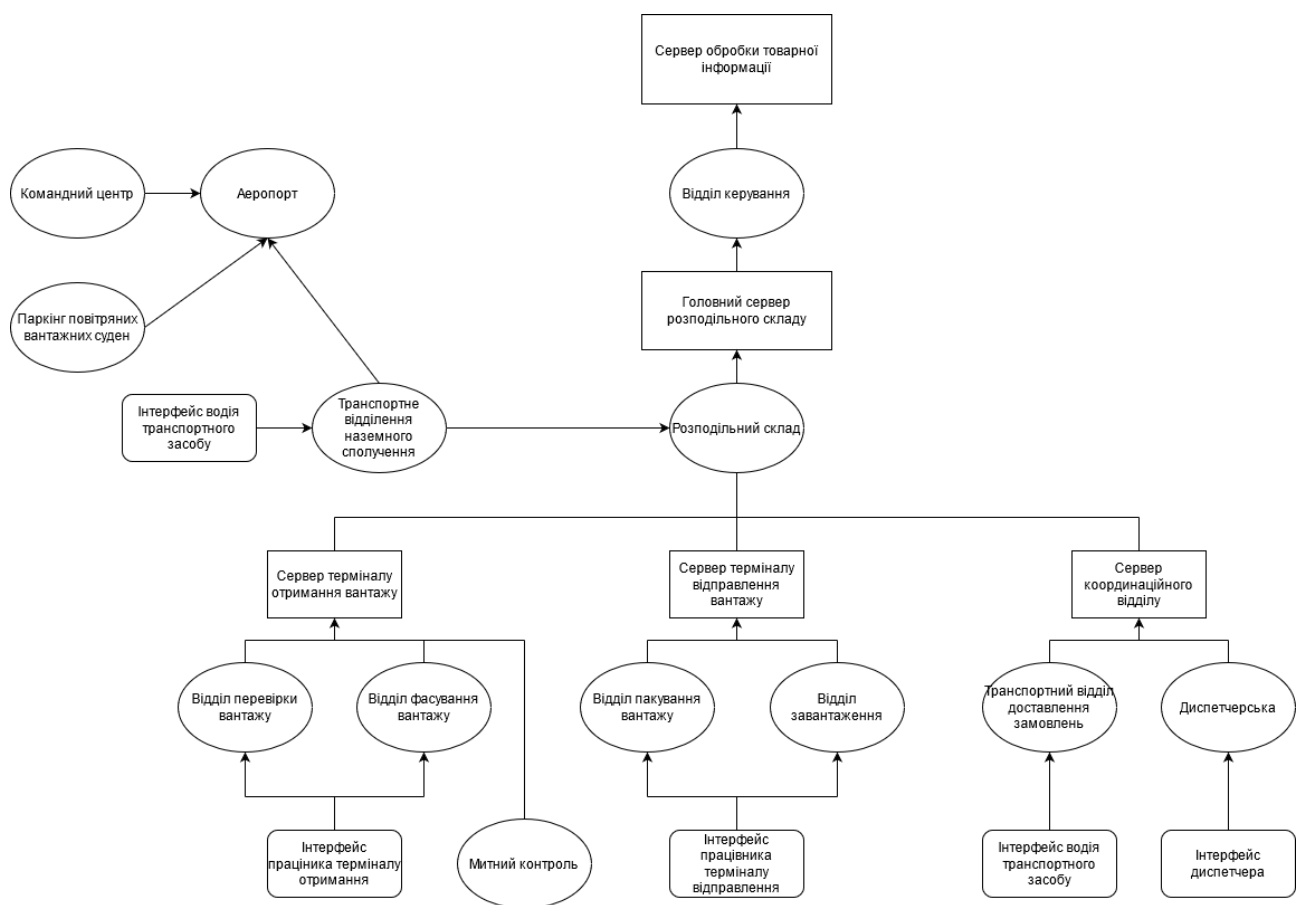


Рис. 2.1. Загальне відображення структури взаємодії інтерфейсів системи логістичної компанії.

Увесь процес доставки вантажу виглядає наступним чином. Компанія однієї країни потребує певного товару, який виробляє фірма іншої країни. Вона робить відповідну заявку на придбання товару у необхідній кількості. Далі фірма виробник

передає товар до локальної логістичної фірми, яка доставляє наземним транспортом до найближчого аеропорту, який здатний прийняти літак з необхідною вантажопід'ємністю. Потім повітряним простором вантаж потрапляє до аеропорту країни, в якій знаходиться компанія замовник. Аеропорт в свою чергу оснащений відповідним терміналом та обладнанням, за допомогою якого можна швидко забрати вантаж з повітряного судна та дати дозвіл на посадку іншому. Надалі вантаж з терміналу аеропорту наземним транспортом перевозиться до терміналу прийому вантажу розподільного складу логістичної компанії. Кожне замовлення підлягає митному контролю. Після інвентаризації та узгодження походження товару, його надають відділу фасування, де товар, який потребує спеціального транспортування поміщується у необхідні контейнери. З відділу фасування увесь вантаж направляють до відділу завантаження де вантажівки транспортного відділу доставлення замовлень отримують вантаж.

Увесь процес знаходиться під контролем відділу керування логістичної компанії. Кожен працівник з вище вказаних відділів під час роботи взаємодіє зі спеціальним інтерфейсом, через який головний сервер обробки товарної інформації отримує необхідні дані для забезпечення інформацією усі компоненти системи.

Одну з головних ролей виконує інтерфейс водіїв вантажних транспортних засобів логістичної компанії. За допомогою якого вони надають та оновлюють поточний стан доставки вантажу до кінцевого споживача, компанії замовника. Кожен інтерфейс системи повинен мати раціональну структуру, яка забезпечить комфортне використання співробітником компанії, покращить його продуктивність та пришвидшить процес доставлення вантажу, тим самим збільшить товарообіг компанії. Що є ключовим фактором економічного розвитку.

Метою роботи є покращити інтерфейс водіїв вантажних наземних транспортних засобів. Приклад раціональної структури інтерфейсу наведено у наступному параграфі.



## 2.2. Загальна структура інтерфейсу веб сервісу водіїв децентралізованої логістичної компанії. Призначення та основні функції

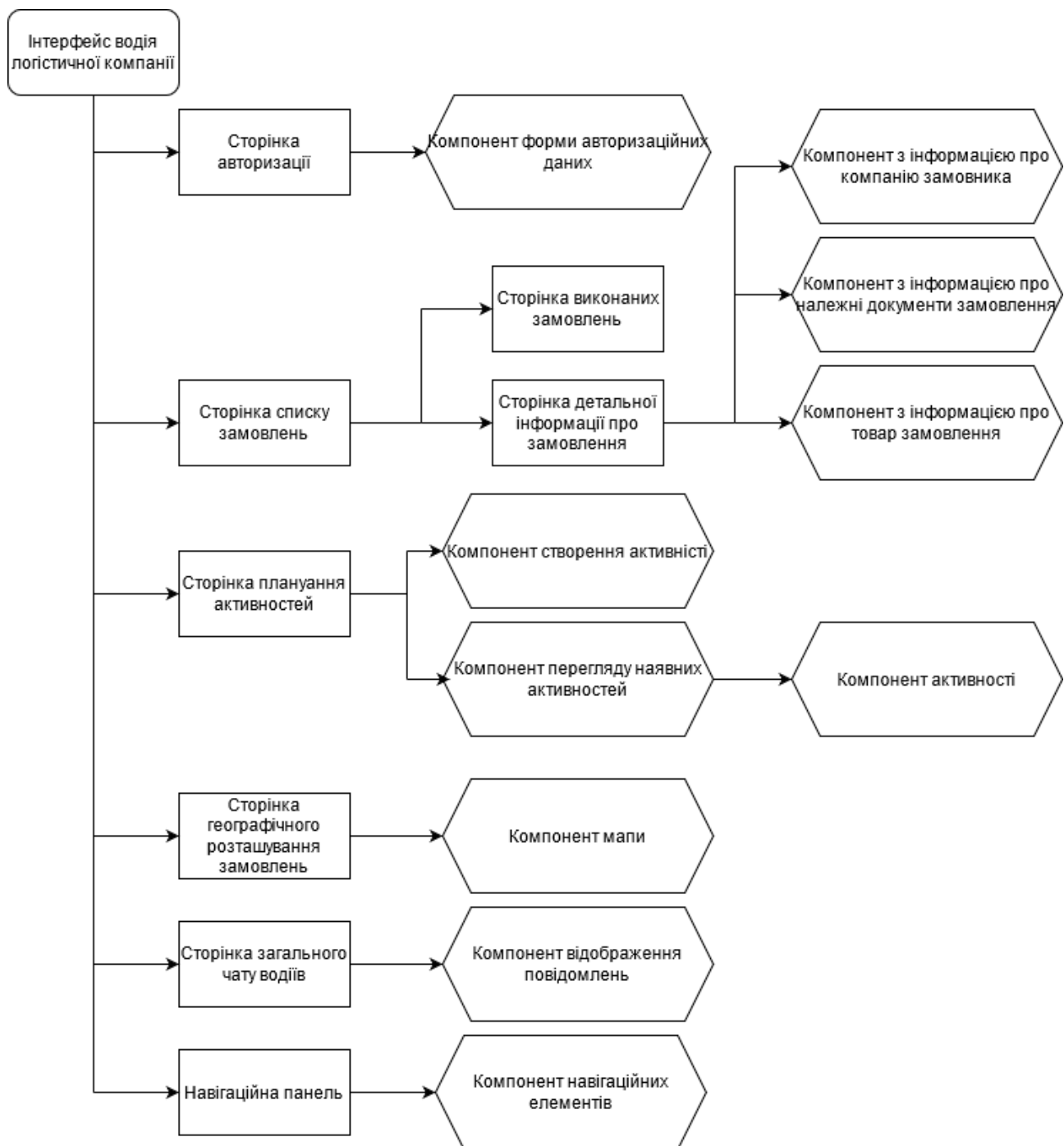


Рис. 2.2. Загальна структура інтерфейсу водіїв вантажних наземних транспортних засобів.

На мою думку подібна організація компонентів інтерфейсу для водіїв логістичної компанії надасть більш точну та актуальну інформацію з приводу того,

скільки замовлень доставляється за певний проміжок часу, які водії належним чином виконують робочі обов'язки, а які порушують умови зазначені у трудовому договорі. Головною ідеєю є повний контроль часових затрат на виконання замовлення, щоб побудувати відповідну політику для покращення логістичних маршрутів та економічних відносин. Також мати можливість надавати оперативні та якісні послуги, направлені на перевезення вантажу будь-якого типу.

### 2.3. Діаграма атрибутів класів елементів інтерфейсу

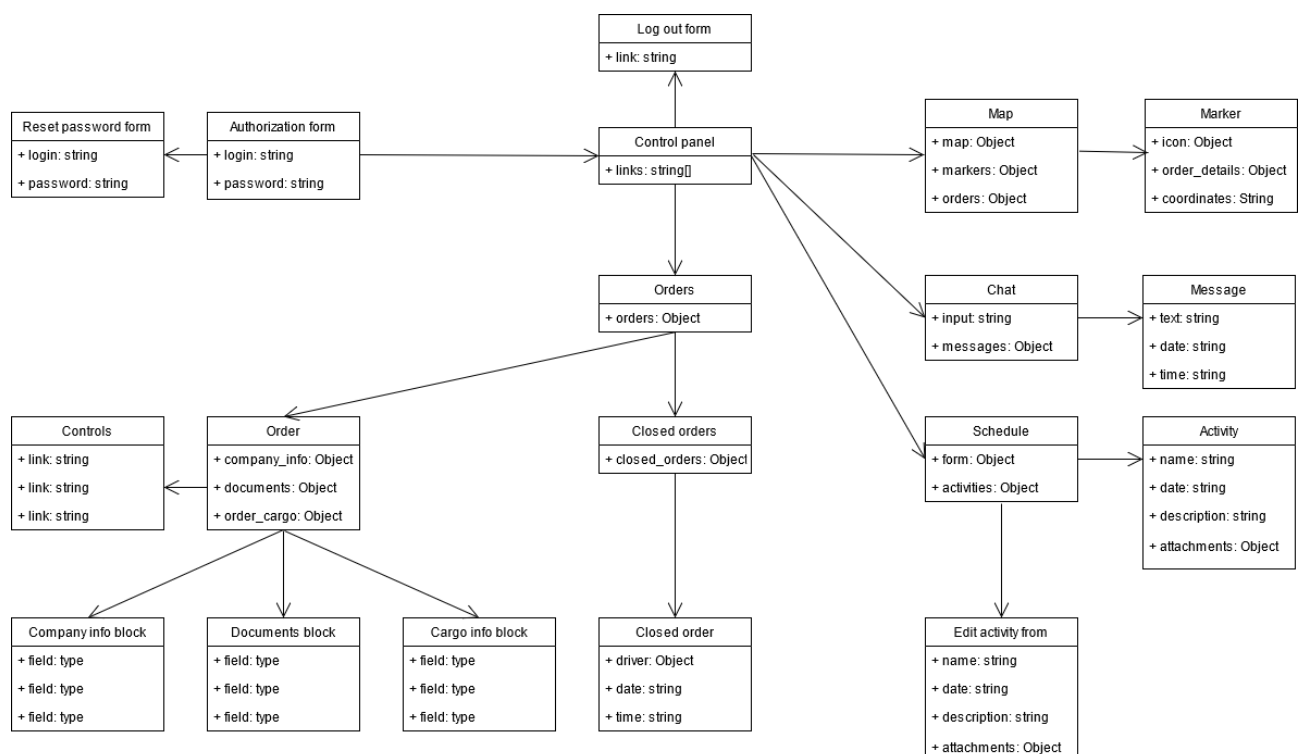


Рис. 2.3. Діаграма атрибутів класів елементів інтерфейсу.

Перш за все, щоб почати взаємодіяти із системою користувачеві необхідно авторизуватися (**Authorization form**), ідентифікувати себе та отримати у власне розпорядження відповідний функціонал. У тому випадку, якщо за якихось обставин втратив пароль для авторизації, інтерфейс надає йому можливість скористуватися формою відновлення паролю (**Reset Password Form**). Після успішної авторизації користувач отримує доступ до навігаційної панелі (**Control panel**), яка супроводжує його серед усіх основних компонентів інтерфейсу. Компонент, який відображає

список усіх замовлень (**Orders**), які необхідно виконати водієві за відведені проміжки часу. Замовлення у списку надають лише загальну інформацію. Для отримання більш детальної інформації стосовно замовлення необхідно обрати його у списку. При натисканні користувач потрапляє на сторінку замовлення (**Order**), вона складається з трьох основних блоків, які надають вичерпну інформацію стосовно поточного замовлення: блок про інформацію компанії замовника, опис пункту прийому (**Company info block**), блок з нормативними документами необхідних для транспортування даного вантажу (**Documents block**), та блок, який надає чіткий опис товару (**Cargo info block**). Також у загальному списку замовлень у кожного замовлення є свої елементи контролю (**Controls**), які надають детальну інформацію, позначають, що замовлення активне, тобто водій прийняв його та готується до відправки та елемент контролю, що свідчить про виконання замовлення. Коли водій виконує замовлення, тобто вантаж доставлено до кінцевої точки, до кінцевого споживача, водій переносить поточне замовлення до списку виконаних (**Closed orders**). Цей список відображає інформацію про виконані водієм замовлення (**Closed order**) впродовж визначеного компанією часом та повідомляє центральній системі, які замовлення виконані, що їх можна переносити з центральної бази до архіву та мати в базі лише не виконані замовлення і відсортувати їх за відповідними пріоритетами.

Також у водія є доступ до компоненту з мапою (**Map**), який візуально відображає географічне взаєморозташування усіх замовлень, тож водій самостійно може визначити пріоритетний та оптимальний шлях, впродовж якого він виконає усі замовлення. На мапі розташовані маркери (**Marker**) по натисканню на який водій отримує загальну інформацію про замовлення та може перейти на детальну сторінку замовлення щоб позначити його, як активне.

Кожен з водіїв має доступ до загального чату (**Chat**), де вони можуть ділитися актуальною інформацією про те, що відбувається на дорогах, повідомлять один одного (**Message**) про вимушені ремонтні роботи або про інциденти на дорозі, щоб інші водії перебудували відповідним чином свій маршрут та не витрачали власний час та гроші компанії очікуючи в заторах.

У кожного водія у розпорядженні є компонент, який надає можливість створити детальний план на робочий день (**Schedule**), тобто сформувати перелік необхідних активностей (**Activity**), які необхідно виконати впродовж дня, також даний компонент може використовуватися, як журнал, тобто коли водій зупинився на відпочинок, він повинен записати дану активність до журналу та компанія буде вести чіткий облік того, як водії розпоряджаються часом і чи належним чином вони виконують свою роботу. Також передбачена можливість редагування створеної активності, якщо водій з певних причин помилився та надав неправильні дані (**Edit activity form**).

## 2.4. Огляд методів та алгоритмів створення інтерфейсу водіїв

Інтерфейс створено на основі веб-сервісу, найкращий вибір для розробки є мова програмування JavaScript, адже вона забезпечує зручну інтеграцію з браузерним середовищем та надає перелік функцій, які значно поліпшують процес розробки і покращують досвід використання цільовим користувачем.

Проте в JavaScript також існують власні недоліки. Як відомо, структура кожної веб-сторінки являє собою так зване DOM(Document Object Model) – дерево, модель об'єкту документа сторінки. DOM дерево формується на основі HTML тегів, які є будівними блоками документу. HTML надає лише статичне відображення сторінки, тобто користувач може лише переглядати інформацію і не має ніякої можливості вплинути на неї. Саме в цей момент на допомогу приходить JavaScript, будь-які дії на сторінці: відправка форми реєстрації, будь-яка кнопка, кожен елемент, який при натисканні спричиняє певні зміни на сторінці, усе це виконується за допомогою JavaScript. JavaScript на пряму взаємодіє з документом веб-сторінки та має повний контроль над кожним елементом сторінки. Це надає можливість створювати надзвичайно гнучкі та комплексні інтерфейси. Проте у всього своя ціна, кожна зміна елементу чи отримання нових даних потребують оновлення сторінки, оновлення DOM-дерева, що є затратною операцією відносно до ресурсів пристрою на якому використовується інтерфейс. На сьогоднішній день, техніка досягла

великого рівня потужності в математичних обчисленнях, на яких базується будь-який програмний продукт, тому якщо справа стосується невеликих додатків, то помітної різниці у швидкодії не спостерігається. Проте, якщо ми хочемо створити інтерфейс наприклад, інтернет-магазину, або будь-якого іншого крупного веб-сервісу, який за короткий проміжок часу надає та приймає величезну кількість інформації, яка може змінюватися не те що кожную хвилину, а кожную секунду. Проблеми з швидкодією будуть помітні, що сповільнить усі процеси, якими оперує система, користувачі будуть вимушені чекати певний час виконання кожної операції.

#### 2.4.1. Алгоритм оптимізації швидкодії інтерфейсу бібліотекою React

Одна з найкрупніших компаній світу Facebook, вирішила знайти відповідне рішення стосовно проблеми швидкодії та створила спеціальну бібліотеку під назвою React

В основі React закладено спеціальний алгоритм, так званий **Reconciliation Algorithm (алгоритм звірки)**, який забезпечує зміну лише того елемента, який це потребує, а не повністю усього дерева. React влючає в себе оптимізації, які створюють уявлення повного відтворення, не зачіпаючи величезну частку продуктивності. Більшу частину цих оптимізацій і влючає в себе процес, званий звірка. Це алгоритм, за яким стоїть те, що називають «Virtual DOM». Визначення звучить яось так: коли ви відмальовується React додаток, дерево елементів, яке описує додаток генерується в зарезервованій пам'яті. Це дерево потім влючається в відображення оточення - на прикладі браузерних додатків, воно перекладається в набір DOM операцій. Коли стан додатку оновлюється (зазвичай викликом спеціальної функції `setState`), нове дерево генерується. Нове дерево порівнюється з попереднім, щоб прорахувати і влючити саме ті операції, які потрібні для перемальовування оновленого додатку.

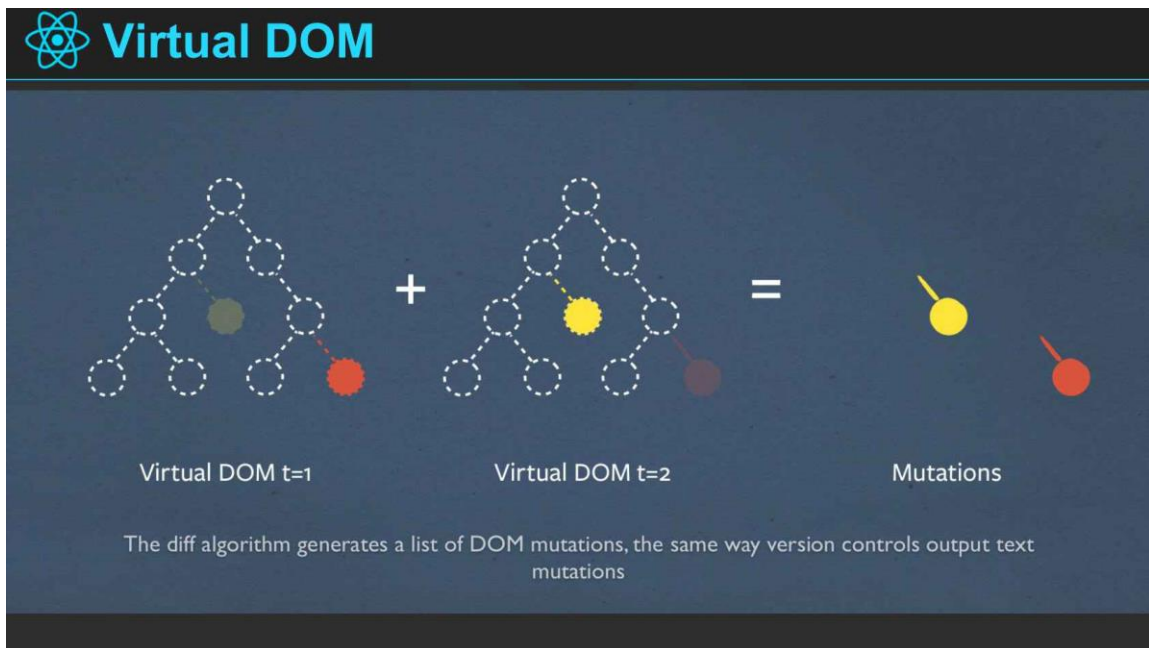


Рис. 2.4. Графічне відображення алгоритму зв'ірки.

Основні переваги алгоритму зв'ірки:

- часті маніпуляції з DOM дорогі та важкі;
- віртуальний DOM - це віртуальне представлення реального DOM;
- коли відбуваються зміни стану, віртуальна DOM оновлюється та порівнюється попередня та поточна версія віртуальної DOM. Це називається "різним";
- потім віртуальний DOM надсилає пакетне оновлення до реального DOM для оновлення інтерфейсу користувача;
- React використовує віртуальний DOM для підвищення своєї продуктивності;
- він використовує спостережуване для виявлення змін стану та опори;
- React використовує ефективний алгоритм diff для порівняння версій віртуального DOM;
- потім він гарантує, що пакетні оновлення надсилаються до реального DOM для перебудування або повторного відтворення інтерфейсу користувача.

## **2.4.2. Алгоритм формування ієрархії даних за допомогою бібліотеки Redux**

Redux - це передбачуваний контейнер стану, розроблений, щоб допомогти писати програми мовою JavaScript, які поведуться послідовно в клієнтському, серверному та рідному середовищах і які легко перевірити.

Хоча він в основному використовується як інструмент управління станом з React, також можна використовувати його з будь-якою іншою структурою JavaScript або бібліотекою. Це невелика вага в 2 КБ (включаючи залежності), тому не доведеться турбуватися про те, що розмір активу програми буде більшим.

За допомогою Redux стан програми зберігається в сховищі, і кожен компонент може отримати доступ до будь-якого стану, який йому потрібен, із цього сховища. Керування станом - це, по суті, спосіб полегшити спілкування та обмін даними між компонентами. Він створює відчутну структуру даних, яка представляє стан програми, з якої можна читати та писати. Таким чином, можна бачити інакше невидимі стани, працюючи з ними.

Більшість бібліотек, таких як React, Angular тощо, побудовані таким чином, щоб компоненти могли внутрішньо керувати своїм станом без потреби у зовнішній бібліотеці чи інструменті. Це добре для додатків з невеликою кількістю компонентів, але в міру того як додаток стає більшим, управління станами, спільними між компонентами, стає складною роботою.

У програмі, де дані обмінюються між компонентами, дуже легко заплутатися та втратити розуміння які данні до яких компонентів належать. В ідеалі, дані в компоненті повинні жити лише в одному компоненті, тому обмін даними між компонентами важкий.

Наприклад, у React, щоб обмінюватися даними між сумісними компонентами, стан повинен знаходитися в батьківському компоненті. Метод оновлення цього стану надається батьківським компонентом і передається як властивість цим компонентам.

Зрозуміло, що управління загальним станом додатку стає важким, оскільки додаток ускладнюється. Ось чому нам потрібен інструмент управління станом, такий як Redux, який полегшує підтримку цих станів усіх компонентів.

Існує центральне сховище станів, яке зберігає весь стан програми. Кожен компонент може отримати доступ до збереженого стану без необхідності пересилати властивості з одного компонента на інший. Є три будівельні частини: дії, сховище та редуктори.

Дії (Actions) є простими об'єктами JavaScript, і вони повинні мати властивість `type`, щоб вказати тип дії, яку потрібно виконати. Вони також повинні мати корисне навантаження, що містить інформацію, над якою слід опрацювати дію.

Редуктори (Reducers) - це чисті функції, які приймають поточний стан програми, виконують дію і повертають новий стан. Ці стани зберігаються як об'єкти, і вони визначають, як змінюється стан програми у відповідь на дію, надіслану до сховища. Як чисті функції, вони не змінюють дані в переданому їм об'єкті та не виконують будь-яких побічних ефектів у додатку. Враховуючи один і той же об'єкт, вони завжди повинні давати однакові результати.

Сховище(Store) зберігає стан програми. Можна отримати доступ до збереженого стану, оновити стан та зареєструвати або скасувати реєстрацію слухачів за допомогою допоміжних методів (`dispatch`).



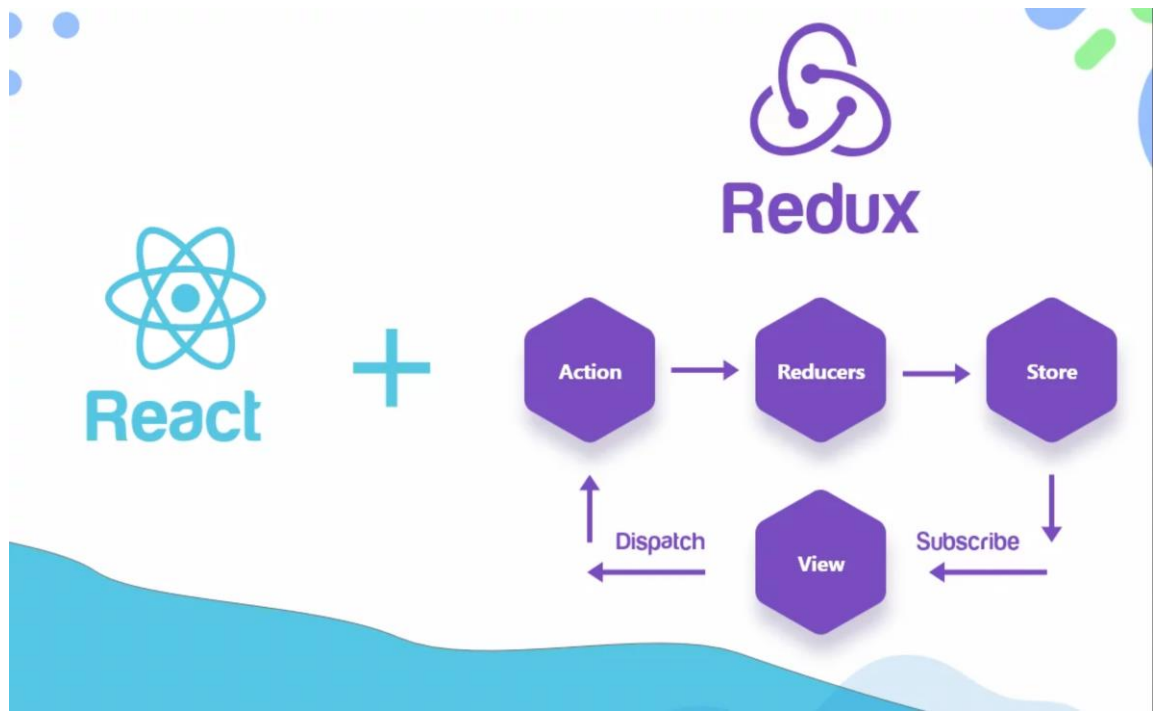


Рис. 2.5. Схема оновлення стану додатку за допомогою Redux.

Redux спрощує налагодження програми. Реєструючи дії та стан, легко зрозуміти помилки кодування, помилки мережі та інші форми помилок, які можуть виникати під час розробки

Окрім того Redux має чудові інструменти розробника (Redux DevTools), які дозволяють вам переміщатись у часі, зберігати дії при оновленні сторінки тощо. Для середніх та великих програм налагодження займає більше часу, ніж розробка функцій. Redux DevTools дозволяє легко скористатися всіма перевагами Redux.

### 2.4.3. Метод покращення взаємодії з React за допомогою NextJs

Щоб створити повну веб-програму за допомогою React з нуля, потрібно врахувати багато важливих деталей:

- код повинен бути зв'язаний за допомогою пакета, такого як webpack, і перетворений за допомогою компілятора, такого як Babel;
- потрібно зробити оптимізацію розробки, наприклад, розбиття коду;
- можливо необхідно статично попередньо відтворити деякі сторінки для продуктивності та SEO;

- можливо доведеться написати якийсь серверний код, щоб підключити програму React до сховища даних.

Фреймворк може вирішити ці проблеми. Але такий фреймворк повинен мати належний рівень абстракції - інакше це не буде дуже корисним. Він також повинен мати чудовий "досвід розробника", що гарантує, що ви та ваша команда маєте вражаючий досвід під час написання коду.

Next.js надає рішення всіх вищезазначених проблем. Next.js має найкращий у своєму класі досвід для розробників та безліч вбудованих функцій:

- інтуїтивна система маршрутизації на основі сторінок (з підтримкою динамічних маршрутів);
- попередній рендеринг, як статичне покоління (SSG), так і рендеринг на стороні сервера (SSR) підтримуються на сторінці;
- автоматичне розділення коду для швидшого завантаження сторінки;
- клієнтська маршрутизація з оптимізованим попереднім вибором;
- вбудована підтримка CSS та Sass, а також підтримка будь-якої бібліотеки CSS-in-JS;
- середовище розробки з підтримкою швидкого оновлення;
- маршрути API для створення кінцевих точок API з безсерверними функціями.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ПРОЕКТУ

#### 3.1. Встановлення залежностей проекту

Перш за все необхідно встановити середовище розробки веб інтерфейсу, а саме NodeJs. Увесь процес розробки відбувається в межах операційної системи Linux з встановленим дистрибутивом Ubuntu 20.0.4. Для того щоб встановити необхідну програму можна скористатися системним інструментом apt (Advanced Packaging Tool). Встановивши NodeJs, в терміналі ми отримали доступ до npm (Node Package Manager), який дозволяє керувати внутрішніми пакетами середовища розробки. За допомогою npm встановив yarn, інструмент, який розроблений компанією FaceBook, яка також є творцем бібліотеки React, яка для покращення взаємодії та мініфікації виникнення проблем сумісності створила власний інструмент для оперування залежностями – yarn. За допомогою yarn встановив спеціальний інструмент create-next-app, який допомагає створити базову структуру проекту та автоматично налаштовує залежності за нас, тим самим зберігає час розробника.

Кожен проект має зарезервовані папки та файли, які відповідають за конфігурацію та взаємодію усіх компонентів. Усі залежності, які додаються в ході розробки зберігаються у спеціальній папці node\_modules, а інформація про ці залежності у спеціальному файлі package.json, який надає перелік абсолютно усіх залежностей проекту. За допомогою цього файлу немає необхідності зберігати та передавати усі залежності. Достатньо передати цей файл та розробник за допомогою спеціальної команди, яка зчитає інформацію з цього файлу автоматично встановить усі необхідні залежності, щоб запустити проект.

|                  |                |  |  |                     |         |  |       |         |  |  |
|------------------|----------------|--|--|---------------------|---------|--|-------|---------|--|--|
| Кафедра КІТ (47) |                |  |  | НАУ 21 14 92 000 ПЗ |         |  |       |         |  |  |
| Виконав          | Коровін Д.О.   |  |  | РЕАЛІЗАЦІЯ ПРОЕКТУ  | Літера  |  | Аркуш | Аркушів |  |  |
| Керівник         | Зіатдінов Ю.К. |  |  |                     |         |  | 35    | 54      |  |  |
| Консульт.        |                |  |  |                     | 411 122 |  |       |         |  |  |
| Н-котрол.        | Шевченко О.П.  |  |  |                     |         |  |       |         |  |  |
| Зав. каф.        | Савченко А.С.  |  |  |                     |         |  |       |         |  |  |
|                  |                |  |  |                     |         |  |       |         |  |  |

```
{ } package.json X
{} package.json > ...
3  "version": "0.1.0",
4  "private": true,
5  "scripts": {
6    "dev": "next dev",
7    "build": "next build",
8    "start": "next start"
9  },
10 "dependencies": {
11   "@date-io/date-fns": "1.x",
12   "@date-io/moment": "1.x",
13   "@emotion/react": "11.1.5",
14   "@emotion/styled": "11.1.5",
15   "@material-ui/core": "^5.0.0-alpha.27",
16   "@material-ui/icons": "^4.11.2",
17   "@material-ui/lab": "^5.0.0-alpha.27",
18   "@material-ui/pickers": "^3.2.10",
19   "@material-ui/styles": "^4.11.3",
20   "@reduxjs/toolkit": "^1.5.1",
21   "date-fns": "^2.19.0",
22   "firebase": "^8.4.3",
23   "google-maps-react": "^2.0.6",
24   "moment": "^2.29.1",
25   "next": "10.0.7",
26   "next-redux-wrapper": "^6.0.2",
27   "react": "17.0.1",
28   "react-chat-elements": "^10.14.0",
29   "react-dom": "17.0.1",
30   "react-dropzone": "^11.3.2",
31   "react-firebase-hooks": "^3.0.4",
32   "react-google-maps": "^9.4.5",
33   "react-icons": "^4.2.0",
34   "react-redux": "^7.2.3",
35   "react-responsive-carousel": "^3.2.15",
36   "redux": "^4.0.5",
37   "redux-batched-actions": "^0.5.0",
38   "redux-devtools-extension": "^2.13.9",
39   "redux-thunk": "^2.3.0",
40   "styled-components": "^5.2.1",
41   "sweetalert2": "^11.0.0"
42 },
43 "devDependencies": {
44   "babel-plugin-inline-react-svg": "^2.0.1",
45   "babel-plugin-styled-components": "^1.12.0",
46   "babel-plugin-transform-require-ignore": "^0.1.1",
47   "install": "^0.13.0"
48 }
49 }
```

Рис. 3.1. Вміст файлу залежностей проекту package.json.

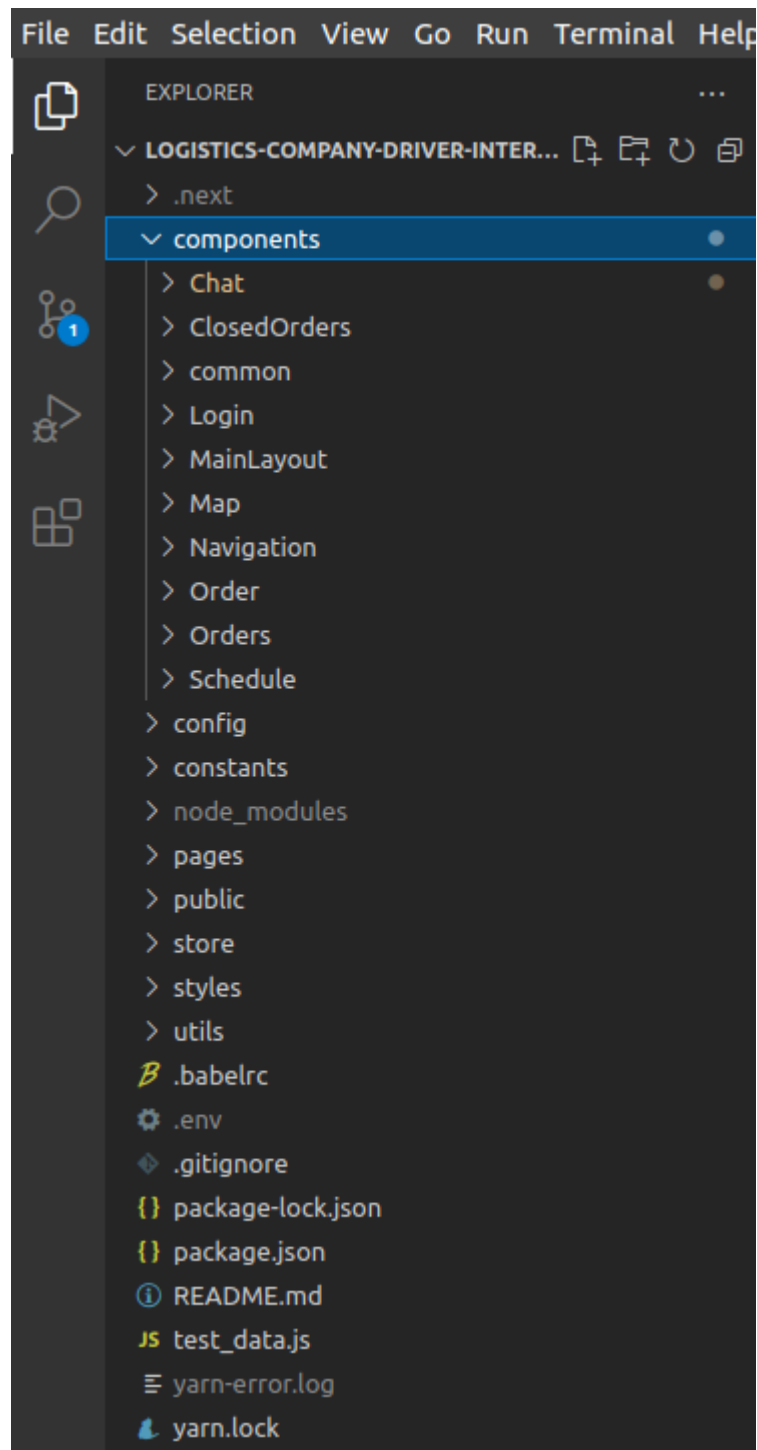
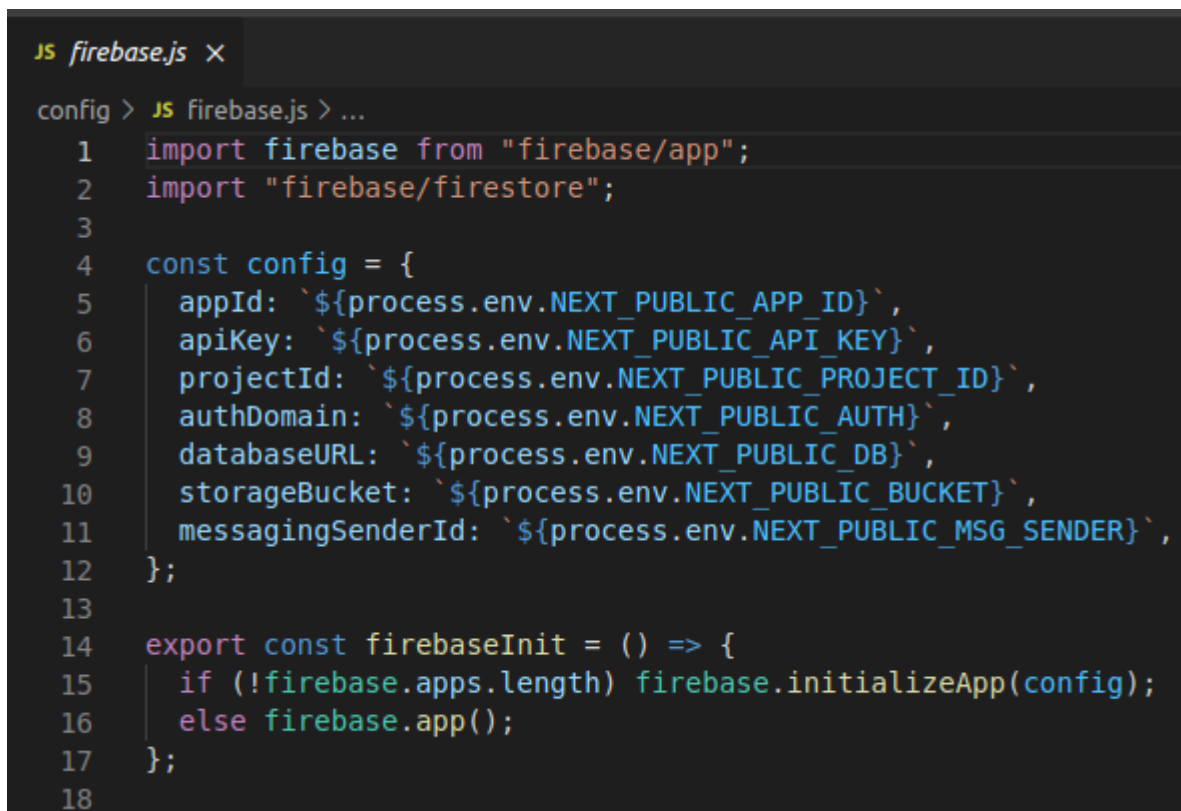


Рис. 3.2. Структура проекту.

Папка `pages` є зарезервованою, усі файли в ній відповідають за навігацію між компонентами інтерфейсу. Кожен файл у цій папці оброблює компоненти, які описує розробник. Прийнято усі компоненти створювати у кореневій папці `components`, проте вона не є зарезервованою.

Для коректного функціонування інтерфейсу необхідно налаштувати сервер, який буде посередником між клієнтом інтерфейсу та базою даних і безлічі інших можливих сервісів, як поштові повідомлення, засоби аналітики використання та інші. Оскільки основною темою роботи є саме інтерфейс та його структура тому для демонстративних цілей було обрано сервіс Firebase від компанії Google, який надає ряд необхідного функціоналу, а саме : базу даних для збереження текстової інформації та базу даних для збереження файлів. Також даний сервіс має спеціальний пакет firebase, який можливо встановити, як залежність до мого проекту. За допомогою якого можливо комфортно взаємодіяти з усім функціоналом. Для цього на сайті сервісу треба провести необхідні налаштування. Перш за все – ініціалізувати проект, кожному проекту присвоюється спеціальний ключ доступу, який надає можливість користуватися сервісом та підтверджує статус авторизованого користувача і запобігає несанкціонованій діяльності. Даний ключ та інші дані, які описують проект необхідно надати до об'єкту конфігурації для пакету firebase.



```
JS firebase.js x
config > JS firebase.js > ...
1  import firebase from "firebase/app";
2  import "firebase/firestore";
3
4  const config = {
5    appId: `${process.env.NEXT_PUBLIC_APP_ID}`,
6    apiKey: `${process.env.NEXT_PUBLIC_API_KEY}`,
7    projectId: `${process.env.NEXT_PUBLIC_PROJECT_ID}`,
8    authDomain: `${process.env.NEXT_PUBLIC_AUTH}`,
9    databaseURL: `${process.env.NEXT_PUBLIC_DB}`,
10   storageBucket: `${process.env.NEXT_PUBLIC_BUCKET}`,
11   messagingSenderId: `${process.env.NEXT_PUBLIC_MSG_SENDER}`,
12 };
13
14 export const firebaseInit = () => {
15   if (!firebase.apps.length) firebase.initializeApp(config);
16   else firebase.app();
17 };
18
```

Рис. 3.3. Об'єкт конфігурації пакету firebase.

Оскільки дані полів цього об'єкту є конфіденційною інформацією компанії, яка використовує сервіс або розробника то їх значення зберігаються у спеціальному файлі середовища .env. Функція firebaseInit перевіряє чи існує вже сутність пакету в даному проєкті, якщо так то відбувається запуск, якщо ні – ініціалізація.

### 3.2. Створення колекцій бази даних

На сайті сервісу Firebase необхідно перейти до розділу Firestore Database та створити основні колекції бази даних.

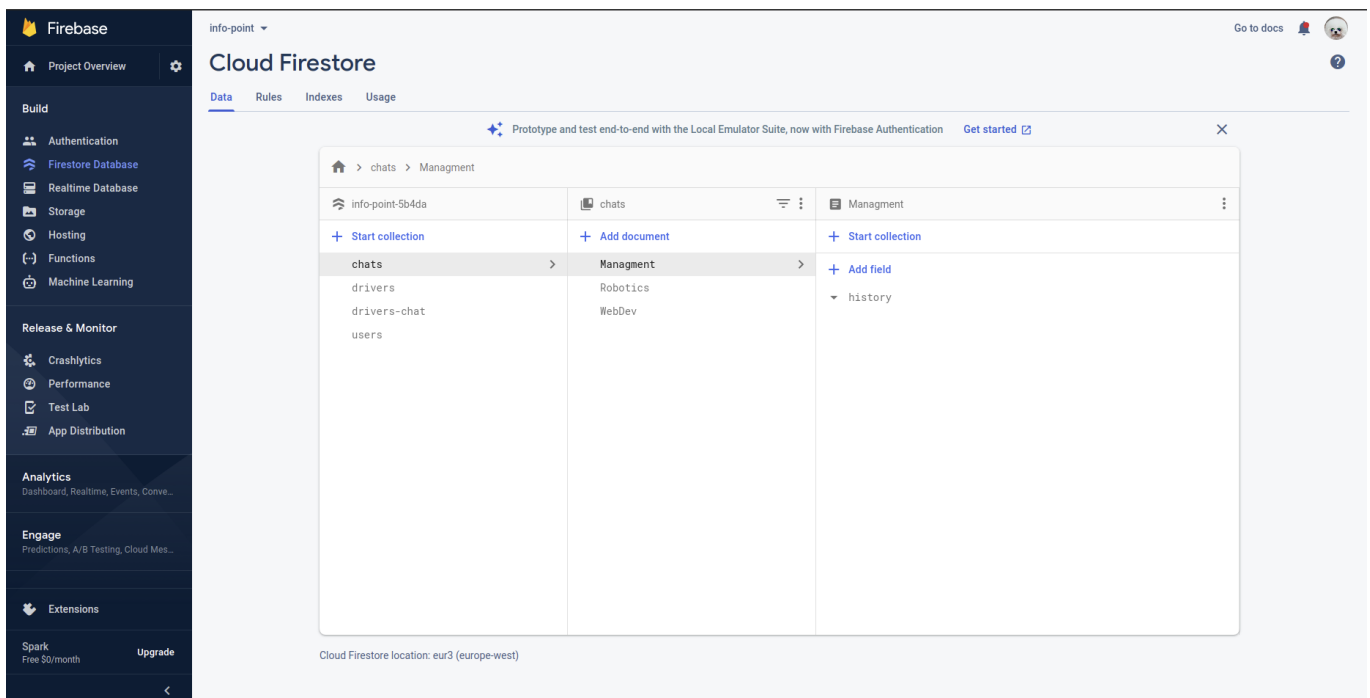


Рис. 3.4. Розділ сервісу Firebase Database.

Перша колонка описує головні колекції бази даних, друга – документи колекції, третя – поля, які описують документ колекції, в свою чергу поле може бути іншою колекцією.

Інтерфейс включає в себе чат для водіїв та надає можливість обміну медіа зображеннями. Для збереження медіа файлів необхідно налаштувати розділ Firebase під назвою Storage. Є можливість створити окремі папки для групування тематичних

медіа файлів або для сортування файлів за типом: фото, аудіо чи відео. Створюю папку driver-chat-photos в яку будуть зберігатися усі фото надані водіями до чату.

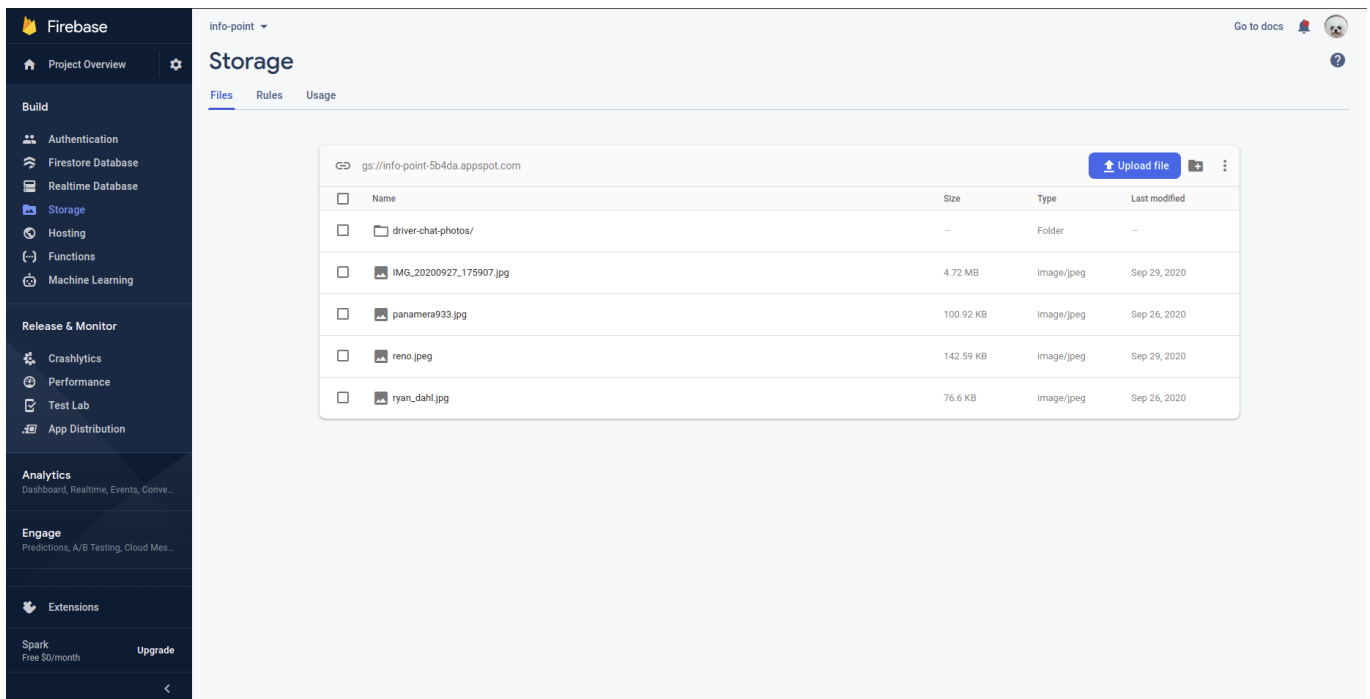


Рис. 3.5. Коренева структура розділу Storage.

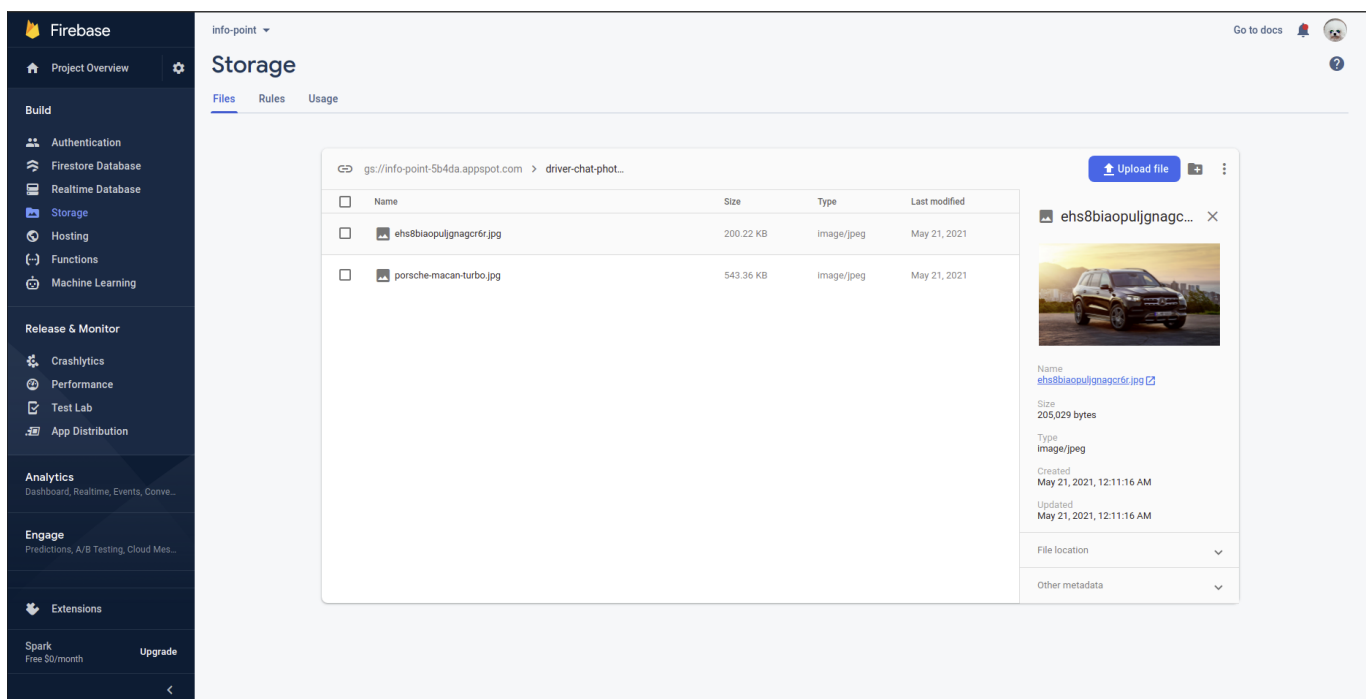


Рис. 3.6. Вміст папки driver-chat-photos.



Виділивши файл можна отримати певні дані стосовно нього: дату та час створення, розмір файлу, посилання на файл та інші.

### 3.3. Налаштування оновлення повідомлень чату в режимі реального часу

Після монтування компоненту чату у вікні браузера буде викликано спеціальний метод компоненту `useEffect`, який ініціалізує метод за допомогою якого компонент підпишеться на оновлення документів колекції бази даних `onSnapshot`. Цей метод після зміни колекції буде повертати оновлені дані та буде зберігати їх локально за допомогою методу `setMessages` і тим самим оновлювати графічне відображення інтерфейсу.

```
63  const Chat = () => {  
64    const [message, setMessage] = useState("");  
65    const [messages, setMessages] = useState([]);  
66  
67    const classes = useStyles();  
68  
69    useEffect(() => {  
70      const unsubscribe = collection  
71        .orderBy("createdAt")  
72        .onSnapshot((querySnapshot) => {  
73          const data = querySnapshot.docs.map((doc) => ({  
74            ...doc.data(),  
75            id: doc.id,  
76          }));  
77  
78          setMessages(data);  
79        });  
80  
81      return unsubscribe;  
82    }, []);
```

Рис. 3.7. Ініціалізація методу моніторингу оновлення даних колекції.

### 3.4. Основні елементи веб інтерфейсу

В другому розділі дипломної роботи було надано представлення структури компонентів інтерфейсу у вигляді загальної діаграми системи, яке місце займає інтерфейс водіїв в межах усієї системи логістичної компанії та діаграми атрибутів класів, які основні параметри характеризують кожен компонент.

В даному розділі буде надано графічне зображення кожного компоненту та детальний опис процесу його створення та функціонування.

Розпочнемо з початкового компоненту – компоненту авторизації користувача.

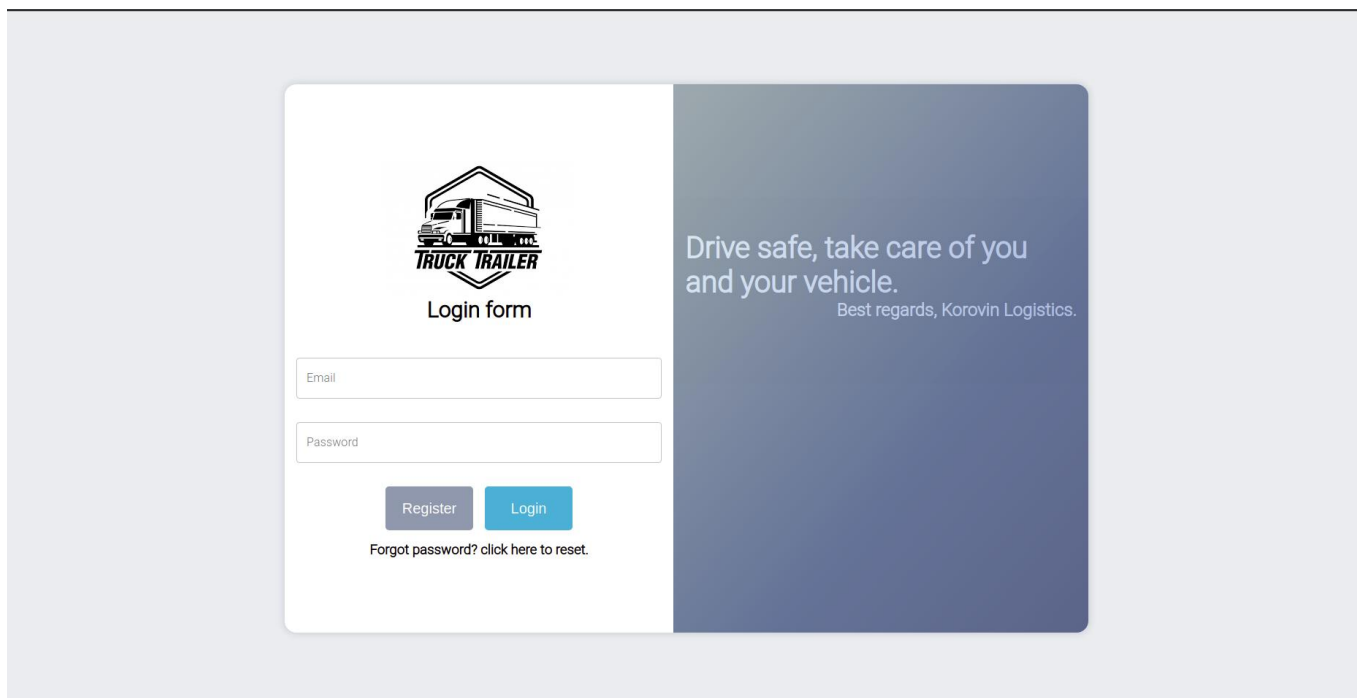


Рис. 3.8. Компонент авторизації користувача.

Кожний додаток, яким користується будь-яка компанія вимагає авторизації користувача, щоб надати йому відповідні привілеї для взаємодії з системою. Логістичні компанії не є винятком.

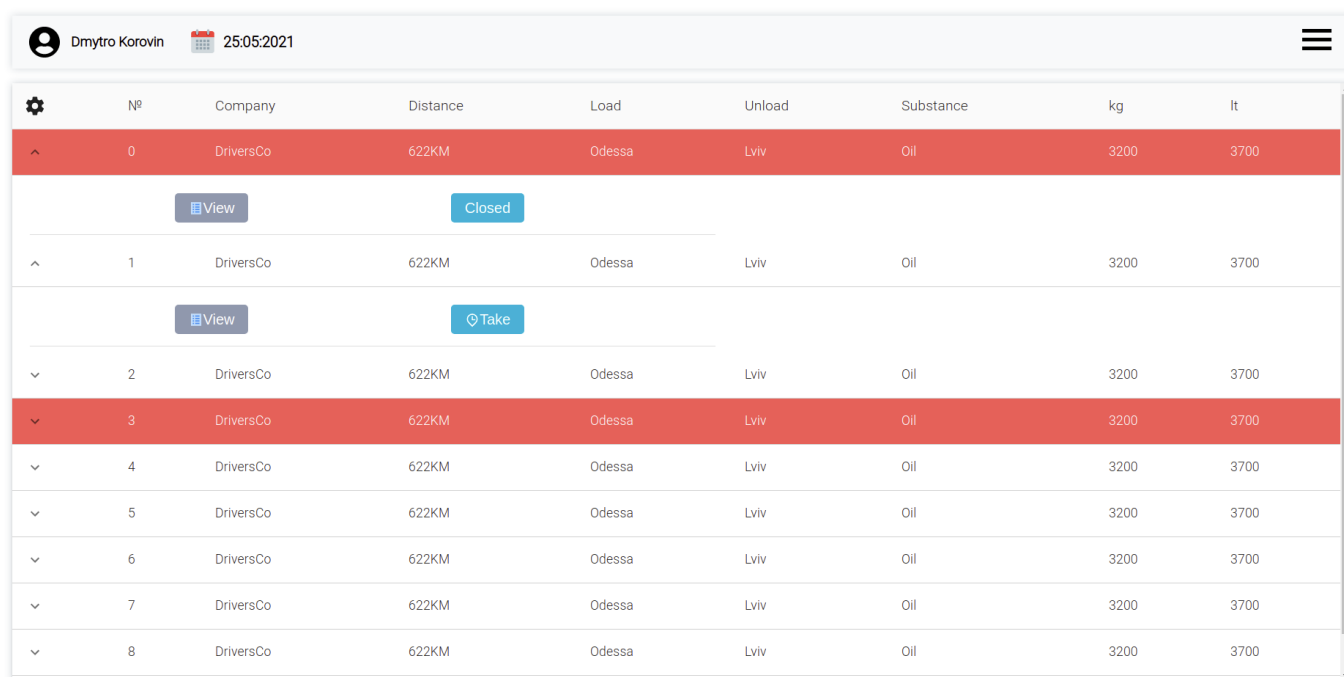
Компонент авторизації блокує доступ до ключових компонентів інтерфейсу доки користувач не введе свої дані, які будуть передані на обробку до серверу та сервер у відповідь надішле спеціальний токен, який буде надалі використовуватися при кожному запиті на сервер та ідентифікувати користувача.

У випадку якщо користувач забув пароль у нього є можливість його змінити: на сервер буде відправлений запит, за яким на пошту користувача надійде лист з інструкціями по встановлення паролю.

### 3.4.1. Компонент замовлень призначених водієві

Головною метою при створенні інтерфейсу було надання чіткої послідовності дій. Для того щоб користувач за короткий час інтуїтивно засвоїв увесь функціонал інтерфейсу та легко запам'ятавав де знаходяться відповідні компоненти.

Після успішної авторизації користувач, а саме водій вантажівки, потрапляє на компонент в якому надано перелік усіх необхідних до виконання замовлень.




| №                      | Company   | Distance | Load   | Unload | Substance | kg   | lt   |
|------------------------|-----------|----------|--------|--------|-----------|------|------|
| 0                      | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| <div>View Closed</div> |           |          |        |        |           |      |      |
| 1                      | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| <div>View Take</div>   |           |          |        |        |           |      |      |
| 2                      | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| 3                      | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| 4                      | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| 5                      | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| 6                      | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| 7                      | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| 8                      | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |


Рис. 3.9. Компонент замовлень призначених водієві.


Компонент являє собою таблицю, яка відображає перелік усіх замовлень водія. Кожен елемент списку надає коротку характеристику про замовлення: його номер в списку, назва компанії замовника, відстань від логістичного центру до замовника, місто де розташована компанія замовника, тип вантажу, його вага та вміст у літрах.


Також кожен елемент має кнопки керування, які надають можливість переглянути усю інформацію стосовно замовлення та перевести замовлення до списку активних.

Детальна інформація складається з трьох розділів: уся інформація про компанії замовника, скани необхідних нормативних документів та детальний опис вантажу.

 Dmytro Korovin

 26:05:2021






CUSTOMER INFO

INFORMATION

SUBSTANCE INFO



| Location     |                | Contacts                 |            | Terminal        |            |
|--------------|----------------|--------------------------|------------|-----------------|------------|
| Distance     | 622KM          | Additional phone         | 0287459874 | All trucks      | ✓          |
| Postal code  | 10030          | Contact person name      | Fedor      | Parking lot     | ✗          |
| City         | Lviv           | Contact persone phone    | 0879654123 | Corona measures | ✓          |
| Country      | Ukraine        | Position                 | manager    | Security point  | ✓          |
| Company name | DriversCo      | Administration           | 0447842468 | Entrie point    | main gates |
| Address      | Lukich str. 4A | Contact person last name | Checkirin  | Equipment       | ✓          |

Рис. 3.10. Детальна інформація про компанію замовника.

Надається повний опис локації куди необхідно доставити вантаж, повна адреса пункту прийому. Контактна інформація: ім'я та контактні дані відповідальних за прийом осіб зі сторони замовника. Опис пункту прийому: чи здатний термінал прийняти усі типи вантажівок, наявність місця паркування, чи в сучасних умовах дотримуються заходи запобігання вірусу та інші.

Dmytro Korovin

26:05:2021

CUSTOMER INFO
INFORMATION
SUBSTANCE INFO

Take

### SOFTWARE DOCUMENTATION TYPES

```

graph TD
    PD[Project documentation] --> Prod[Product documentation]
    PD --> User[User documentation]
    PD --> Process[Process documentation]
    Prod --> Sys[System documentation]
    Prod --> End[End-users documentation]
    User --> Admin[System admin documentation]
    Sys --> SysList["✓ Product requirement document  
✓ Design and architecture  
✓ Agile product roadmaps  
✓ Source code document  
✓ User experience design documentation  
✓ Testing documents  
✓ Help and maintenance"]
    End --> EndList["✓ Plans  
✓ Estimates  
✓ Schedules  
✓ Reports and metrics  
✓ Working papers  
✓ Standards"]
    Admin --> AdminList["✓ Plans  
✓ Estimates  
✓ Schedules  
✓ Reports and metrics  
✓ Working papers  
✓ Standards"]

```

1 of 3

Рис. 3.11. Перелік нормативних документів.

Dmytro Korovin

26:05:2021

CUSTOMER INFO
INFORMATION
SUBSTANCE INFO

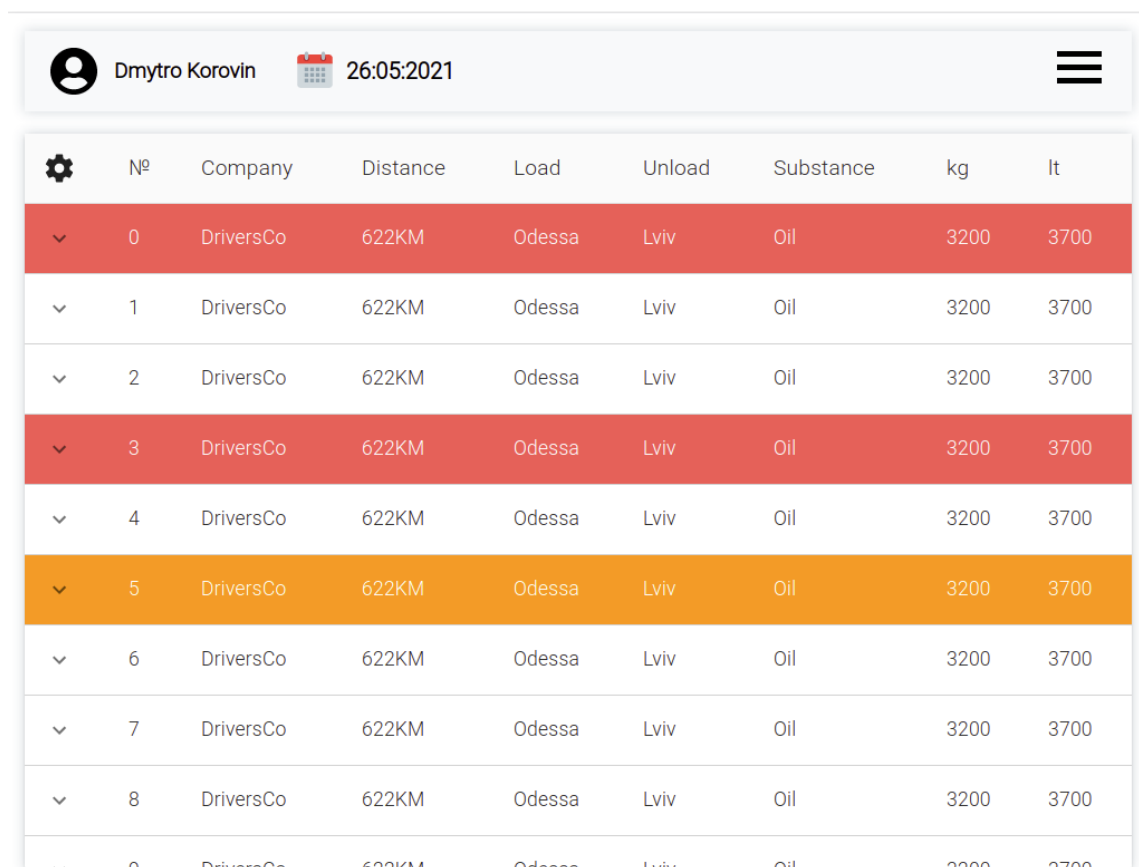
Take

|                                     |   |
|-------------------------------------|---|
| Weight kg                           | 3200  |
| Weight lt                           | 3700  |
| <div> </div> <div>Description</div> | <p>During normal use, impurities such as dirt, metal scrapings, water, or chemicals can get mixed in with the oil, so that in time the oil no longer performs well. Eventually, this used oil must be replaced with virgin or re-refined oil to do the job at hand. EPA's used oil management standards include a three-pronged approach to determine if a substance meets the definition of used oil. To meet EPA's definition of used oil, a substance must meet each of the following three criteria:</p> <p>Origin - Used oil must have been refined from crude oil or made from synthetic materials. Use - Oils that are used as lubricants, hydraulic fluids, heat transfer fluids, buoyants, and for other similar purposes are considered used oil. Unused oils such as bottom clean-out waste from virgin fuel oil storage tanks or virgin fuel oil recovered from a spill, do not meet EPA's definition of used oil because these oils have never been "used." EPA's definition also excludes products used as cleaning agents or used solely for their solvent properties, as well as certain petroleum-derived products like antifreeze and kerosene. Contaminants - In other words, to meet EPA's definition, used oil must become contaminated as a result of being used. This aspect of EPA's definition includes residues and contaminants generated from handling, storing, and processing used oil. Physical contaminants could include metal shavings, sawdust, or dirt. Chemical contaminants could include solvents, halogens, or saltwater.</p> |
| Name                                | Oil   |

A basic table example with a caption

Рис. 3.11. Детальна інформація про вантаж.

Якщо перемістити замовлення до списку активних то у загальному списку воно змінить колір на жовтий, тобто замовлення очікує завершення.



|   |   |           |          |        |        |           |      |      |
|---|---|-----------|----------|--------|--------|-----------|------|------|
| ⚙ | № | Company   | Distance | Load   | Unload | Substance | kg   | lt   |
| ▼ | 0 | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| ▼ | 1 | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| ▼ | 2 | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| ▼ | 3 | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| ▼ | 4 | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| ▼ | 5 | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| ▼ | 6 | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| ▼ | 7 | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| ▼ | 8 | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |
| ▼ | 9 | DriversCo | 622KM    | Odessa | Lviv   | Oil       | 3200 | 3700 |

Рис. 3.12. Зміна статусу замовлення.

Для того, щоб переглянути список активних замовлень необхідно скористатися боковою панеллю навігації, яка забезпечує переміщення між усіма компонентами веб інтерфейсу. На мою думку саме таке розташування основного елементу керування є найраціональнішим, адже уся увага користувача буде сконцентрована безпосередньо на поточному компоненті та інформації, яку він надає, не відволікаючи увагу сторонніми елементами.

Також над кожним компонентом інтерфейсу знаходиться інформаційна панель, яка відображає ім'я поточного користувача, поточну дату та кнопку виклику навігаційної панелі.

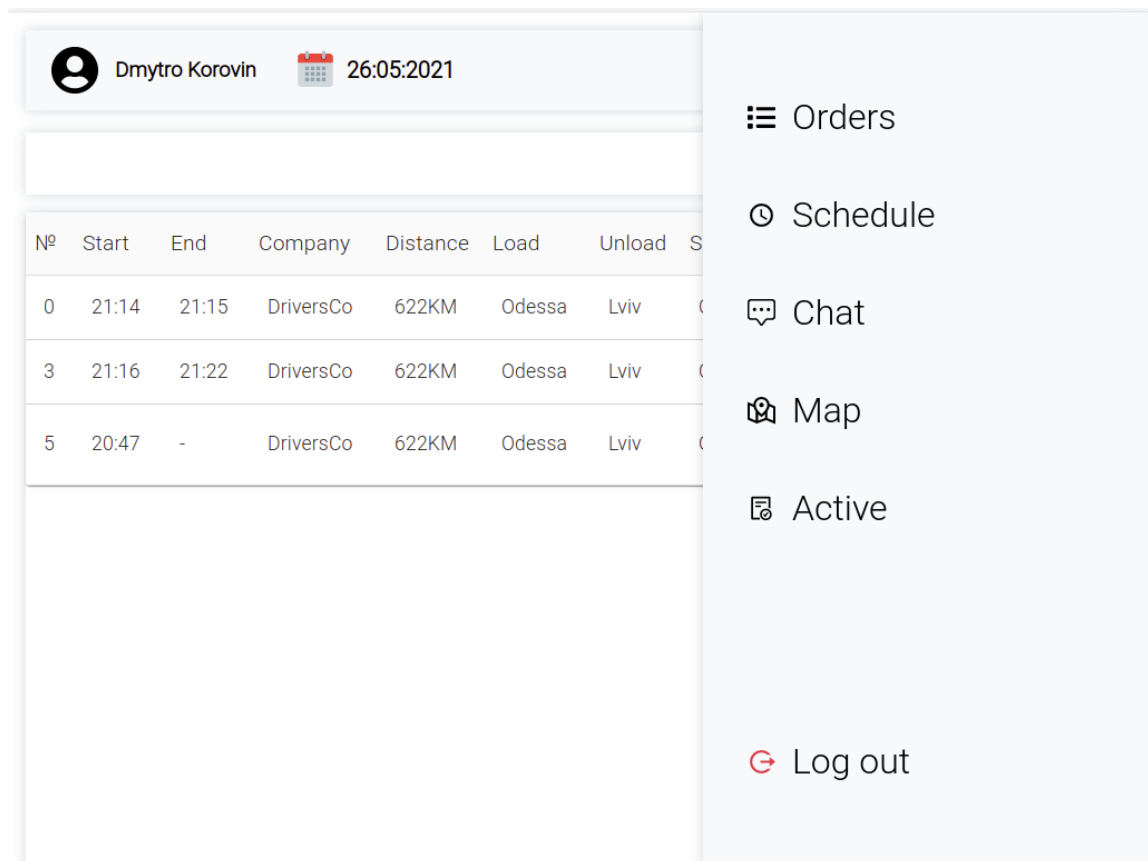


Рис. 3.13. Інформаційна та навігаційна панель інтерфейсу.

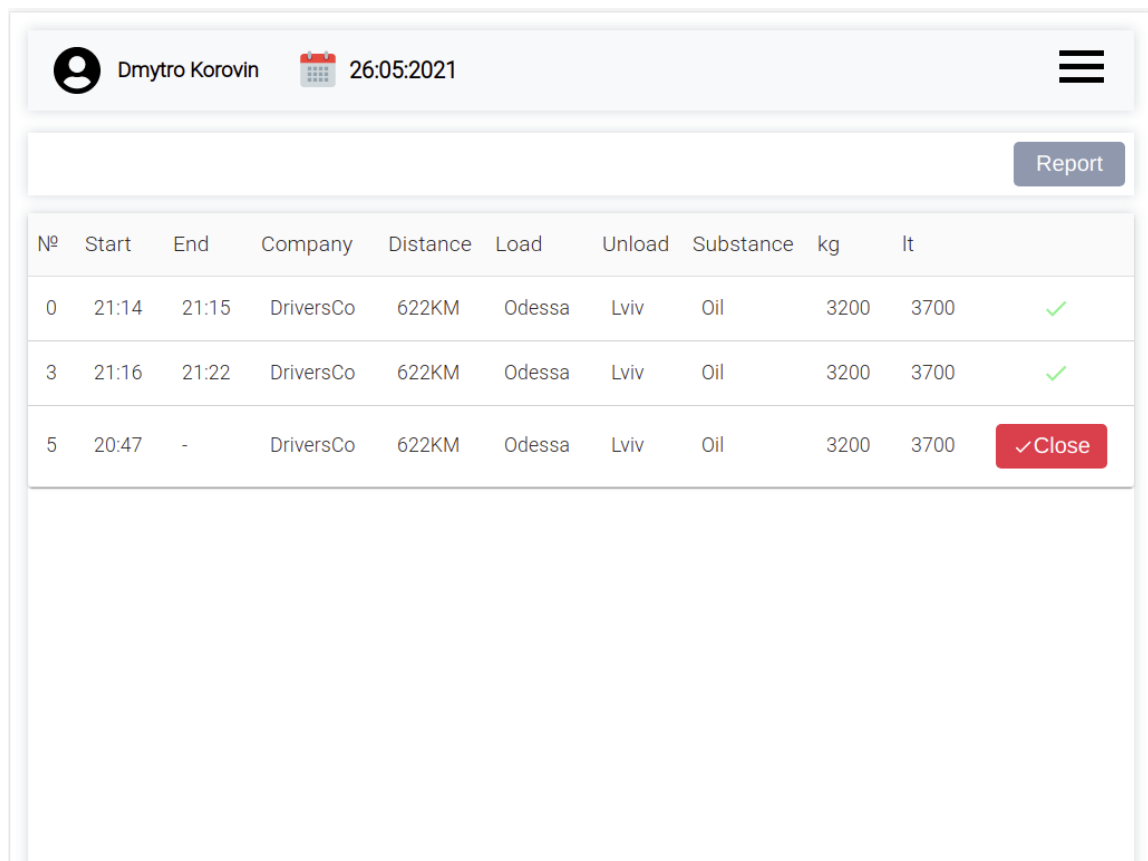


Рис. 3.14. Список активних замовлень.

Після доставки замовлення водієві необхідно закрити його. Після закриття замовлення у списку активних замовлення отримує помітку виконаного, а в загальному списку змінює колір на червоний та отримує статус виконаного.

### **3.4.2. Компонент ведення план-графіку діяльності водія**

Кожен водій вантажівки в день проводить понад десяти годин в дорозі. Враховуючи, що він людина та має необхідні людські потреби у вигляді сну, туалету та прийомів їжі, йому необхідно робити зупинки. Також, враховуючи жвавість дорожнього трафіку у дорозі не виключні ситуації, які сповільнюють трафік машин, а саме аварії і від цього ніхто не застрахований. Попри все деякі водії халатно відносяться до виконання своїх прямих обов'язків та витрачають більше часу на відпочинок ніж їм відведено та необхідно. Саме тому, враховуючи усі викладені вище факти, компанії необхідно реалізувати чіткий контроль за діяльністю водіїв у робочий час та отримувати детальний план графік, який буде пояснювати необхідність затримки та надавати точну інформацію про те, скільки часу та на що відводить водій під час рейсу.

Саме тому було вирішено додати компонент планування та обліку часу де водій буде вносити нотатки та інформувати компанію де і скільки часу він витратив впродовж дня. Компонент складається з двох частин: перша – форма введення даних: часу, дати, назви діяльності та опису цієї діяльності, друга – список створених нотаток. Також передбачена можливість редагування створеного запису та можливість завантаження фото, яке буде асоціюватися з даною діяльністю. Наприклад: компанія видає водієві певну суму грошей, які він повинен витратити на їжу, номер в готелі або при необхідності в дорозі провести технічний огляд автомобіля. Водієві необхідно створити нотатку, що о такій годині, такого числа він снідав у кафе та надати фотокопію або скріншот виписки з банку при розрахунку за чеком. Таким чином компанія знатиме скільки годин з відведених водій витрачає на відпочинок так скільки коштів він витрачає.



The screenshot shows a user interface for a driver's schedule. At the top, there is a header bar with a user profile icon, the name "Dmytro Korovin", a calendar icon, the date "26:05:2021", and a hamburger menu icon. Below the header, on the left, is a form for adding a new note. It includes a "SELECT TIME" section with a digital clock showing "09:13" and "AM", and a circular analog clock with the number "9" highlighted. Below the clock is a date input field showing "26-05-2021". Further down are fields for "Activity" (a dropdown), "Activity type" (a dropdown), and a text input for "Enter personal activity". At the bottom of the form are two buttons: "Reset" (red) and "Add note" (blue). On the right side of the interface, there is a large gray area with the text "Schedule is empty..." in a large, light gray font.

Рис. 3.15. Компонент план-графіку водія.

This screenshot shows the same user interface as the previous one, but with a note added to the schedule. The time in the "SELECT TIME" section is now "10:35" and "PM", with the number "10" highlighted on the analog clock. The "Add note" button is now disabled. On the right side, the "Schedule is empty..." text is replaced by a list of notes. The first note is titled "Clean car" and has a description "Take car to the car wash". Below the note, there is a timestamp "26-05-2021 at 10:34". Above the note list, there are three tabs: "ALL NOTES" (which is selected and underlined), "PERSONAL", and "COMPANY". Each note has edit and delete icons (pencil and 'x') to its right.

Рис. 3.16. Створення запису план-графіку.

Рис. 3.17. Вікно редагування запису план-графіку.

### 3.4.3. Компонент мапи

Кожен водій користується навігаторами, пристроями, які здатні урахувати безліч факторів дорожнього руху на поточний період часу. Замість того щоб купувати окремі пристрої навігаційного забезпечення, можна скористуватися додатковими веб сервісами, які надають графічне зображення мапи та дозволяє розробникам програмного забезпечення розширювати його відповідно до конкретних потреб замовника. Для свого інтерфейсу у якості демонстрації було обрано веб сервіс від компанії Google, який має одну з найбільших баз географічних координат Земної кулі. Також дозволяє відображати додаткову інформації поверх локацій на мапі. Також можливо налаштувати навігаційного асистента, але для цього необхідно придбати корпоративну підписку на використання даного сервісу.

На мапі відображається три типи маркерів: водії, маркер місце знаходження компанії куди необхідно доставити вантаж та маркер логістичного центру компанії. Маркери замовлення є інтерактивними. По натисканню на маркер з'являється інформаційне вікно, яке надає загальну інформацію про замовлення та елементи контролю статусом замовлення.

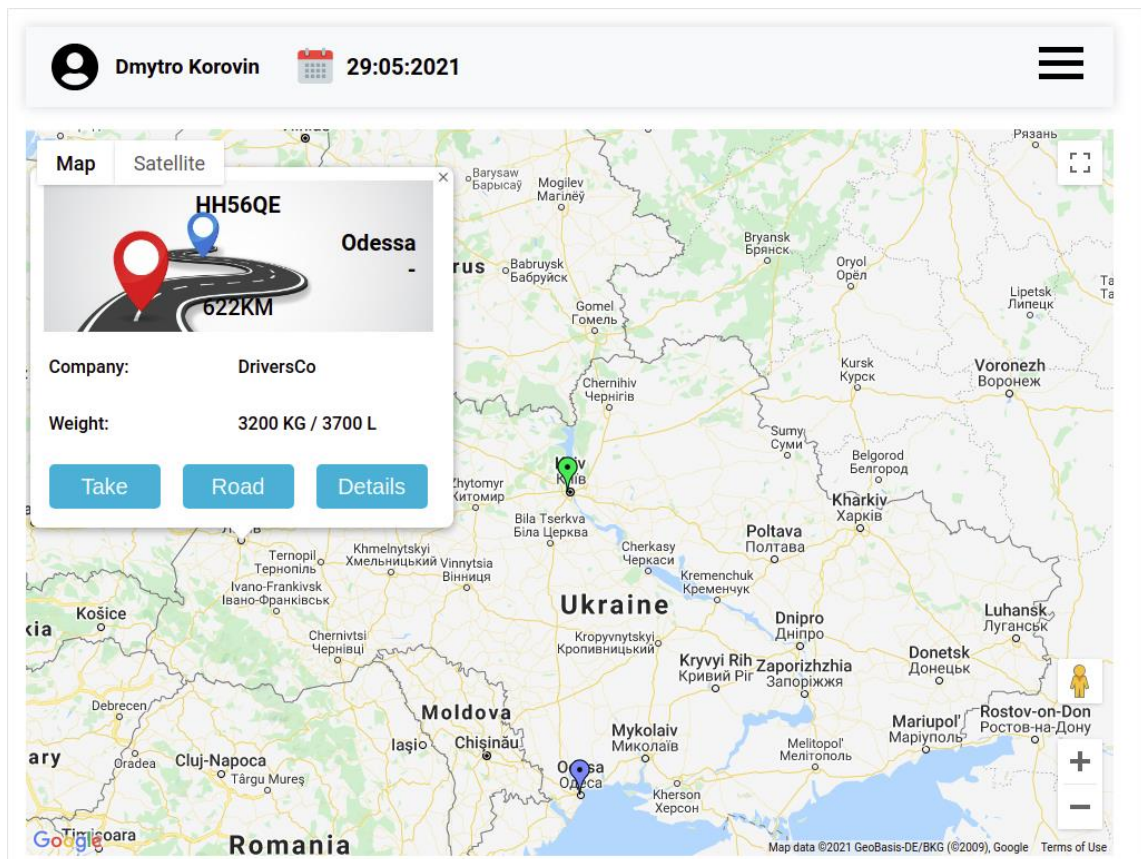


Рис. 3.18. Компонент інтерфейсу – мапа.

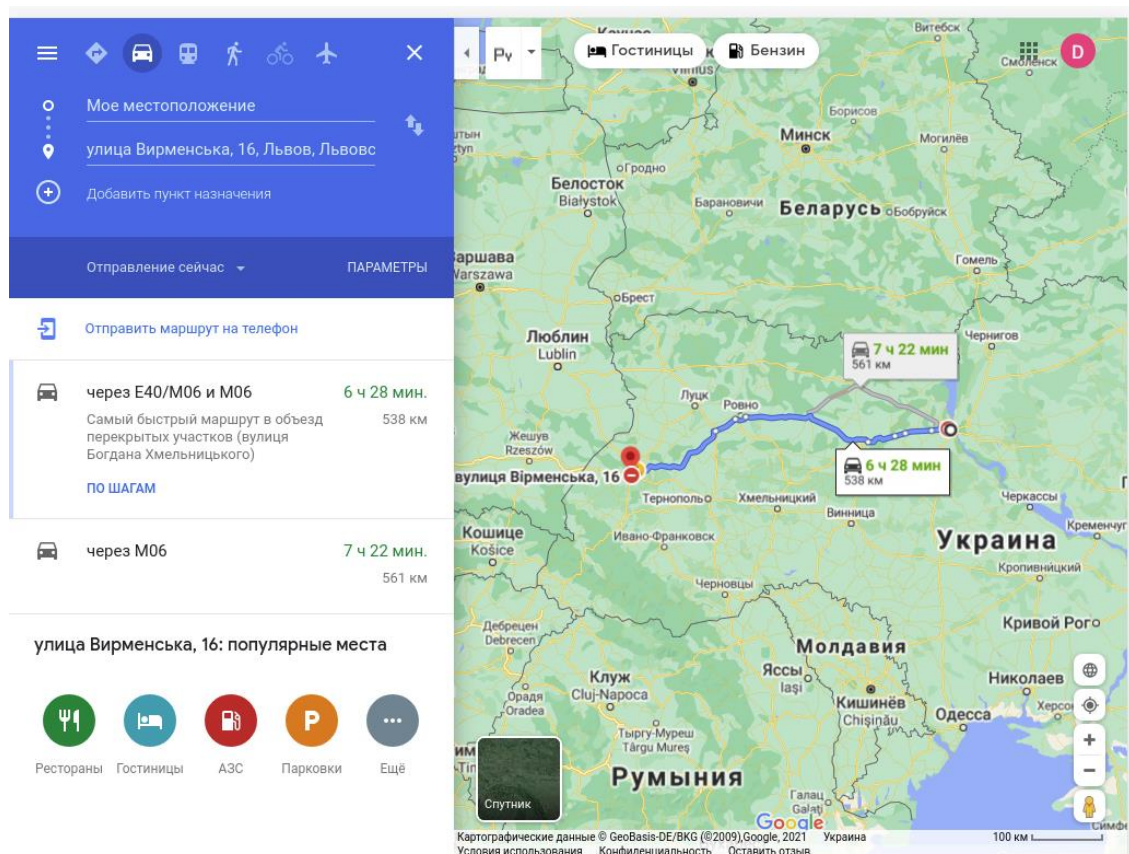


Рис. 3.19. Навігаційний маршрут.

### 3.4.4. Компонент чату

Водії компанії представляють собою певну спільноту, якій необхідно обмінюватися певною інформацією. На даний момент водії користуються раціями. Використання веб інтерфейсу у якості засобу комунікації надає більше варіантів обміну інформацією ніж голосова рація обмеженого радіусу дії. Можливість у реальному часі обмінюватися медіа інформацією такою як фото, відео, аудіо та можливість провести аудіо та відео зв'язок у режимі реального часу.

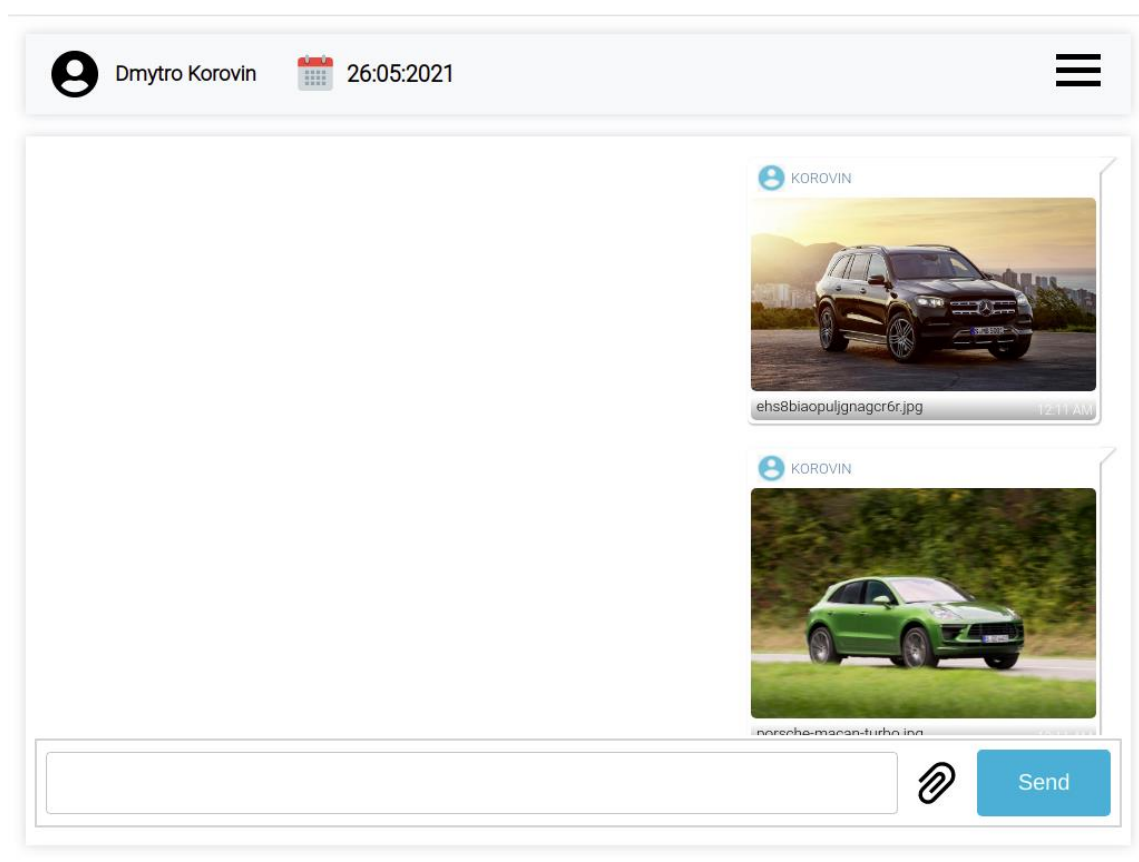


Рис. 3.20. Компонент чату веб інтерфейсу.

## ВИСНОВКИ

Перевагами використання представленого інтерфейсу є його незалежність від операційної системи де він буде використовуватися. Також не є проблемою тип пристроїв, за допомогою яких можна взаємодіяти з даним інтерфейсом адже мова програмування JavaScript у сукупності з прилежними бібліотеками по типу React надає широкий спектр інструментів для створення адаптивних інтерфейсів. Єдиною вимогою до пристрою є наявність браузера в системі, що є виконавчим середовищем програмного коду за допомогою якого створено інтерфейс. Але ця вимога не є критичною, адже JavaScript дуже легко інтегрувати до додатків, які написані на NodeJs (виконавче середовище, що дозволяє виконувати програми написані мовою JavaScript за межами браузера та розширює функціонал даної мови програмування).

Також на сьогоднішній день важко уявити людину, яка не вміє взаємодіяти з веб-додатками. Тому приязність інтерфейсу до користувача та швидкість його освоєння не повинні стати проблемою.

Каналом зв'язку передачі даних виступає глобальна мережа Інтернет, що вже має у власному арсеналі інструменти, що забезпечують безпеку передачі даних, за необхідністю можливо створити власний канал зв'язку, який буде оснащений більш потужним механізмом захисту та кодування даних.

Єдиним недоліком інтерфейсу може бути залежність від стабільного інтернет покриття, що може сповільнити отримання актуальних даних. Проте в умовах сучасного світу подібні логістичні компанії існують в межах великих міст де розвинений великий рівень інфраструктури і подібні проблеми з втратою зв'язку виникають рідко.

## СПИСОК БІБЛЮГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Learn JavaScript. [Електронний ресурс] режим доступу: <https://learn.javascript.ru/> (дата звернення 11.05.2021 р). – Назва з екрана.
2. Node Js. [Електронний ресурс] режим доступу: <https://nodejs.org/uk/> (дата звернення 12.05.20201 р). – Назва з екрана.
3. HTML. [Електронний ресурс] режим доступу <https://www.w3schools.com/html/> (дата звернення 13.05.2021 р). – Назва з екрана.
4. CSS. [Електронний ресурс] режим доступу: <https://www.w3schools.com/css/> (дата звернення 14.05.2021 р). – Назва з екрана.
5. React. [Електронний ресурс] режим доступу: <https://ru.reactjs.org/> (дата звернення 01.06.2021 р). – Назва з екрана.
6. Redux. [Електронний ресурс] режим доступу: <https://redux.js.org/> (дата звернення 03.06.2021 р). – Назва з екрана.
7. NextJs. [Електронний ресурс] режим доступу: <https://nextjs.org/> (дата звернення 08.06.2021 р). – Назва з екрана.
8. Firebase. [Електронний ресурс] режим доступу: <https://firebase.google.com/> (дата звернення 10.05.2021 р). – Назва з екрана.
9. Google Cloud Console. [Електронний ресурс] режим доступу: <https://console.cloud.google.com/> (дата звернення 14.05.2021 р). – Назва з екрана.
10. Yarn [Електронний ресурс] режим доступу: <https://yarnpkg.com/> (дата звернення 13.05.2021 р). – Назва з екрана.
11. Вікіпедія [Електронний ресурс] режим доступу: <https://uk.wikipedia.org/wiki/Логістика/> (дата звернення 11.05.2021 р). – Назва з екрана.